

Étude comparative de serveurs multiterminologiques dans le contexte  
d'une plateforme d'échange de données de santé ontologiquement  
annotées

par

Abdoul Wahab Touré

Mémoire présenté au Département d'informatique  
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES  
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, décembre 2021

Le 09 décembre 2021

*le jury a accepté le mémoire de M. Abdoul Wahab Touré dans sa version finale.*

**Membres de jury**

Professeur Luc LAVOIE  
Directeur de recherche  
Département d'informatique

Professeur Jean-François ÉTHIER  
Évaluateur interne  
Département de médecine

Professeur Nadia TAHIRI  
Présidente rapporteuse  
Département d'informatique

# Sommaire

*Encore aujourd'hui, l'utilisation de systèmes d'information de santé est souvent problématique en raison de l'absence d'un référentiel sémantique et d'une classification unifiée des données des patients. Cette absence est en grande partie attribuable à l'hétérogénéité des sources sur lesquelles reposent ces systèmes. Cette hétérogénéité nuit en outre au partage des données entre les cliniciens, les chercheurs et les administrateurs au sein et entre les réseaux de santé, partage qui serait pourtant bénéfique autant pour la recherche en santé qu'à la pratique de soins. Une première étape dans la résolution de ces problèmes consiste à classer correctement les données en recourant à des terminologies au moyen de serveurs multiterminologiques (SMT). Ces derniers modélisent les terminologies, en plus de fournir une gamme de services permettant de définir et de mettre en correspondance plusieurs terminologies. Dans ce mémoire, nous cherchons un tel serveur qui aurait le potentiel d'être intégré à une démarche de Système de santé apprenant (SSA) et plus particulièrement à la Plateforme apprenante pour la recherche en santé et services sociaux (PARS3) qui propose un service sûr d'accès aux données de santé ontologiquement annotées. Une méthodologie de comparaison des serveurs a été mise en place afin d'évaluer les performances de chacun en termes d'utilisabilité, ainsi que de capacité à définir de nouvelles terminologies propres à des acteurs particuliers. Ce mémoire utilise une comparaison euristique en regard d'un ensemble de critères basés sur les besoins des utilisateurs, les propriétés des serveurs et des artefacts qui leur sont liés. Cette comparaison prend en compte les fonctions accessibles à l'utilisateur par l'interface personne-machine ainsi que les fonctions accessibles à des services par l'entremise d'une interface machine-machine. Il est ressorti de cette comparaison qu'aucun des serveurs étudiés ne répondait adéquatement aux besoins. Cela a néanmoins permis d'identifier les fonctionnalités faisant défaut. Finalement, à partir de ces résultats, il a fallu trancher à savoir s'il était plus profitable de développer un serveur propre ou d'améliorer un serveur existant afin qu'il réponde aux exigences requises. Il en résulte que le développement d'un nouveau serveur serait plus avantageux.*

# Remerciements

Avant tout, je remercie Dieu Le Tout Puissant de m'avoir donné la patience pour réaliser ce travail malgré toutes les difficultés que j'ai rencontrées.

Je tiens à remercier sincèrement toutes les personnes qui ont contribué au succès de ce mémoire.

À leur tête se trouve mon directeur, Professeur Luc Lavoie, à qui j'exprime toute ma reconnaissance, pour sa patience et ses conseils judicieux, qui n'a ménagé aucun effort pour me guider et m'aider, et de qui j'ai beaucoup appris sur tous les plans scientifique, humain et éthique. Je n'ai jamais imaginé dans ma vie de rencontrer un Professeur aussi exceptionnel que Luc sur tous les plans. Merci infiniment.

Je tiens à remercier sincèrement Dominique Wolfshagen pour son aide dans la rédaction de ce mémoire et à Docteur Jean François Ethier pour son soutien moral et intellectuel.

Je voudrais exprimer mes remerciements aux membres de jury.

Et aussi un grand merci à mes très chers parents qui ont toujours été là pour moi et à ma femme, à mes frères, mes sœurs, mes cousins et mes amis pour leurs encouragements.

J'exprime ma reconnaissance envers les amis et tous mes collègues au GRIIS, dont je ne pourrais citer tous les noms, qui m'ont apporté leur soutien moral et intellectuel tout au long de ce mémoire. Que tous ceux et celles qui m'ont aidé et soutenu trouvent ici l'expression de mes remerciements les plus profonds.

En fin, je tiens à remercier le Groupe de recherche interdisciplinaire en informatique de la santé (GRIIS) pour son soutien financier.

# Table des matières

Sommaire .....	iii
Remerciements.....	iv
Table des matières .....	v
Liste des abréviations.....	ix
Liste des figures.....	xi
Liste des tableaux.....	xii
Introduction.....	1
Contexte.....	1
Objectifs .....	3
Méthodologie .....	4
Résultats.....	4
Structure du mémoire .....	5
Chapitre 1 Présentation.....	6
1.1 Mise en contexte.....	6
1.1.1 Systèmes de santé apprenant.....	6
1.1.2 Terminologies.....	10
1.1.3 Serveurs multiterminologiques.....	14
1.2 Problématique.....	16
1.2.1 Rôle du SMT en regard de PARS3 .....	16
1.2.2 Caractéristiques du SMT recherché .....	17
1.3 Question de recherche .....	19
Chapitre 2 Les serveurs multiterminologiques dans les SSA .....	21

2.1	Historique des systèmes terminologiques en médecine .....	21
2.2	Caractérisation des serveurs multiterminologiques.....	22
2.2.1	Rôle et missions.....	22
2.2.2	Méthodes d'accès typiques.....	23
2.2.3	Fonctionnalités usuelles.....	26
2.2.4	Principaux composants et architectures typiques.....	27
2.3	Expression des besoins.....	28
2.3.1	Besoins généraux .....	28
2.3.2	Besoins exprimés par les utilisateurs .....	28
2.4	Fonctionnalités et services.....	31
2.4.1	Définitions des types de ressources principales.....	31
2.4.2	Définition des groupes d'activités principales.....	32
2.4.3	Définition des services .....	32
2.5	Les SMT considérés dans ce mémoire .....	35
2.5.1	SMT intéressants.....	36
2.5.2	Présélection sur la base des besoins identifiés .....	39
<b>Chapitre 3 Méthodes d'évaluation .....</b>		<b>40</b>
3.1	Évaluation de l'utilisabilité.....	40
3.1.1	Processus d'évaluation .....	40
3.2	Évaluation du modèle.....	42
3.3	Évaluation des fonctionnalités des API et des IPM .....	43
3.3.1	Spécifications des exigences des API et des IPM .....	44
3.3.2	Synthèse des critères.....	46
3.4	Évaluation de l'alignement et de la liaison.....	47
3.5	Évaluation de la sécurité .....	49
3.6	Évaluation des autres critères .....	50
<b>Chapitre 4 Étude comparative des serveurs considérés .....</b>		<b>53</b>
4.1	Évaluation de l'utilisabilité.....	53

4.1.1	Documentation.....	53
4.1.2	Facilité de déploiement .....	56
4.2	Évaluation du modèle.....	59
4.2.1	Description du modèle de chaque serveur .....	59
4.3	Évaluation des fonctionnalités des API et des IPM .....	62
4.3.1	Évaluation des API.....	62
4.3.2	Évaluation des IPM.....	67
4.3.3	Synthèse d'évaluation des API et IPM .....	73
4.4	Évaluation de l'alignement et de la liaison.....	74
4.4.1	Liaison sémantique et structurelle.....	74
4.4.2	Alignements .....	75
4.5	Évaluation de la sécurité .....	77
4.5.1	Comparaison des SMT en fonction de la sécurité.....	77
4.6	Évaluation des autres critères .....	80
4.6.1	Contenu.....	80
4.6.2	Architecture et technologies utilisées .....	82
4.6.3	Autres caractéristiques .....	84
4.7	Analyse comparative .....	85
4.7.1	Satisfaction des critères.....	85
4.7.2	Candidat le plus prometteur.....	88
 Chapitre 5 Estimation du projet de développement d'un nouveau SMT ou modification d'un SMT existant.....		 89
5.1	Présentation des méthodes d'estimation utilisées .....	89
5.1.1	Cocomo.....	89
5.1.2	Points de Fonction (FP) .....	90
5.2	Estimation des projets .....	90
5.2.1	Présentation de l'outil d'estimation.....	90
5.2.2	Estimation du développement d'un nouveau SMT .....	92

5.2.3	Faisabilité de la modification de LexEVS .....	104
5.2.4	Conclusion.....	105
	Conclusion .....	106
	Annexe A Modèle FP et Cocomo .....	108
A.1	Modèle FP.....	108
A.1.1	Historique .....	108
A.1.2	Composantes.....	108
A.1.3	Architecture .....	109
A.1.4	Définitions de complexité de chaque composant .....	109
A.1.5	Calcul des points de fonction.....	111
A.1.5.1	Calcul des points de fonction non ajustés (UFP).....	112
A.1.5.2	Calcul des points de fonction ajustés.....	112
A.1.6	Étapes pour compter les points de fonction.....	113
A.2	Cocomo.....	114
A.2.1	Description de la méthode.....	114
A.2.2	Modèle de base.....	115
	Bibliographie .....	116



# Liste des abréviations

API	Application Programming Interface (interface programmatique ou interface de programmation d'application)
CCAM	Classification commune des actes médicaux
CIM	Classification internationale des maladies
Cocomo	Constructive COst Model
CTS2	Common Terminology Services 2
DI	Département d'informatique
FP	Function Points
GRIIS	Groupe de recherche interdisciplinaire en informatique de la santé
HeTOP	Health Terminology/Ontology Portal
IPM	Interface personne-machine
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LexEVS	Lexical Grid Enterprise Vocabulary Services
LOINC	Logical Observation Identifiers Names and Codes
MeSH	Medical Subject Headings
NCI	National Cancer Institute
OBO	Open Biological and Biomedical Ontology
OWL	Web Ontology Language
PAD	Projet d'accès aux données
PARS3	Plateforme apprenante pour la recherche en santé et services sociaux
RDF	Resource Description Framework
REST	Representational State Transfer
SGBD	Système de gestion de base de données
SMT	Serveur multiterminologique
SNOMED CT	Systematized Nomenclature of MEDicine Clinical Terms

SNOMED	Systematized NOMenclature of MEDicine
SOAP	Simple Object Access Protocol
SSA	Système de santé apprenant
T/O	Terminologies et ontologies
UDS	Université de Sherbrooke
XML	eXtensible Markup Language

## Liste des figures

Figure 1 — Cycle d'un SSA [19] .....	7
Figure 2 — Triangle sémiotique .....	11
Figure 3 — Relations existantes entre les terminologies médicales [11] .....	15
Figure 4 — Rôles du SMT en regard de PARS3 .....	17
Figure 5 — Architecture et services du SMT.....	27
Figure 6 — Une partie de l'API REST de LexEVS.....	63
Figure 7 — Une partie de l'API CTS2 de LexEVS. ....	64
Figure 8 — Une partie de l'API Admin de Snow Owl.....	65
Figure 9 — Une partie de l'API REST de BioPortal.....	66
Figure 10 — Utilisation de LexEVS pour une liaison structurelle/sémantique [17] .....	75
Figure 11— Aperçu des résultats du chiffrier .....	91
Figure 12 — Procédure d'estimation.....	92
Figure 13 — Diagramme de classes des terminologies .....	94
Figure 14 — Architecture du modèle des points de fonction [49].....	109
Figure 15 — Procédé de compte des points de fonction.....	114

## Liste des tableaux

Tableau 1 — Synthèse des besoins exprimés par les utilisateurs .....	30
Tableau 2 — Critères d'utilisabilité .....	42
Tableau 3 — Critères d'évaluation du modèle .....	43
Tableau 4 — Critères d'évaluation des API et des IPM .....	47
Tableau 5 — Critères d'évaluation de l'alignement et de la liaison .....	48
Tableau 6 — Critères d'évaluation de la sécurité .....	50
Tableau 7 — Autres critères d'évaluation.....	52
Tableau 8 — Résumé de la documentation de LexEVS.....	54
Tableau 9 — Résumé de la documentation de Snow Owl.....	55
Tableau 10 — Résumé de la documentation de BioPortal .....	55
Tableau 11 — Déploiement pour LexEVS .....	56
Tableau 12 — Déploiement pour Snow Owl.....	57
Tableau 13 — Déploiement pour BioPortal .....	58
Tableau 14 — Synthèse d'évaluation du modèle .....	62
Tableau 15 — Synthèse d'évaluation des API .....	67
Tableau 16 — Évaluation des fonctionnalités d'importation (chargement).....	68
Tableau 17 — Évaluation des fonctionnalités d'exportation .....	68
Tableau 18 — Évaluation des fonctionnalités de consultation d'alignement.....	69
Tableau 19 — Évaluation des fonctionnalités de consultation de SC .....	69
Tableau 20 — Évaluation des fonctionnalités de consultation de CU.....	70
Tableau 21 — Évaluation des fonctionnalités de consultation de JV .....	70
Tableau 22 — Évaluation des fonctionnalités de gestion d'alignement .....	71
Tableau 23 — Évaluation des fonctionnalités de gestion de SC.....	71
Tableau 24 — Évaluation des fonctionnalités de gestion de CU .....	72
Tableau 25 — Évaluation des fonctionnalités de gestion de JV.....	72
Tableau 26 — Synthèse d'évaluation des IPM.....	73

Tableau 27 — Synthèse de l’alignement et de la liaison.....	77
Tableau 28 — Synthèse d’évaluation de la sécurité.....	80
Tableau 29 — Comparaison des contenus.....	82
Tableau 30 — Comparaison de l’architecture et des technologies utilisées .....	84
Tableau 31 — Comparaison des caractéristiques.....	85
Tableau 32 — Comptage des UFP d’EF (Gestion des terminologies).....	95
Tableau 33 — Comptage des UFP d’EF (Import/export de données) .....	96
Tableau 34 — Comptage des UFP d’EF (Gestion des langues).....	96
Tableau 35 — Comptage des UFP d’EF (Gestion des concepts) .....	97
Tableau 36 — Comptage des UFP d’EF (Gestion d’alignement intraterminologique) .....	98
Tableau 37 — Comptage des UFP d’EF (Gestion d’alignement interterminologique) .....	99
Tableau 38 — Comptage des UFP d’EF (Gestion des jeux de valeurs).....	100
Tableau 39 — Comptage des UFP de Données (ILF, EIF) .....	101
Tableau 40 — Nombre de composants et leur niveau de complexité.....	102
Tableau 41 — Estimation de FP non ajustés (UFP).....	102
Tableau 44 — Table de complexité des entrées .....	110
Tableau 45 — Table de complexité des sorties .....	110
Tableau 46 — Table de complexité des requêtes (interrogations).....	110
Tableau 47 — Table de complexité des fichiers internes .....	111
Tableau 48 — Table de complexité des fichiers externes.....	111
Tableau 49 — Table de calcul des points de fonction non ajustés .....	112
Tableau 50 — Estimation des FP ajustés.....	113
Tableau 51 —Types de système .....	115

# Introduction

## Contexte

L'informatisation des dossiers médicaux des patients a permis le développement de nombreux outils contribuant à l'amélioration des soins. Toutefois, jusqu'à présent, l'accès aux données (souvent disséminées en plusieurs dépôts hétérogènes tant par la technologie que par la modélisation) demeure souvent local et contraint par diverses considérations éthiques et légales. La difficulté, voire l'impossibilité, de réunir l'ensemble des données relatives à un patient réduit considérablement la fiabilité et la portée tant des outils cliniques que l'avancement de la recherche, notamment dans la perspective de la mise en place d'un système de santé apprenant (SSA), c'est-à-dire un système accélérant la production et le transfert de connaissances entre les milieux de la recherche et de la pratique de soins. Pour que cette vision puisse se concrétiser, le partage de données doit être possible et efficace, ce qui n'est pas le cas présentement.

En effet, les systèmes de santé sont développés et maintenus par différents intervenants à travers le monde, avec des spécifications ajustées aux besoins des différents pays, régions, villes et ainsi de suite. Par ailleurs, un hôpital n'ayant pas les mêmes besoins qu'un médecin de famille ou un centre local de services communautaire (CLSC), alors de nombreux systèmes d'information différents sont utilisés pour répondre à des besoins spécifiques. En outre, plusieurs acteurs fournissent au domaine médical des logiciels différents [1] qui, le plus souvent, constituent des dépôts de données qui leur sont propres. Toutes ces conditions font en sorte que les systèmes d'information de santé utilisent diverses terminologies pour désigner les mêmes concepts, ce qui mène à une grande hétérogénéité qui fait obstacle à la combinaison de données afin de les rendre interopérables. Le domaine de la santé compte un très grand nombre de terminologies

standardisées, parmi lesquelles Systematized Nomenclature of MEDicine Clinical Terms (SNOMED CT) [2, 3, 4], Classification commune des actes médicaux (CCAM) [5], Classification internationale des maladies, 11<sup>e</sup> version (CIM11) [6], Medical Subject Headings (MeSH) [7, 8], Logical Observation Identifiers Names and Codes (LOINC) [9, 10], et un nombre encore plus grand de terminologies dérivées de celles-ci ou spécifiques à un fournisseur ou un logiciel.

De nombreux serveurs terminologiques ont été développés afin de faciliter l'échange de données entre les systèmes d'information de santé [11]. Un serveur terminologique est un logiciel pouvant héberger, gérer, documenter, diffuser et mettre en correspondance différentes terminologies. Selon Metke-Jimenez et coll. [12], pour qu'un tel serveur soit utilisable en pratique, il doit également comporter un mécanisme de distribution qui facilite la mise à jour des terminologies, tout en ayant des capacités avancées d'interrogation de concepts.

Des chercheurs en science de l'information et en informatique médicale se sont penchés sur le problème de la mise à disposition de serveurs terminologiques dans le domaine de la santé [13]. Dans le monde anglophone, plusieurs serveurs terminologiques ont ainsi été conçus et développés, dont Lexical Grid Enterprise Vocabulary Services (LexEVS), BioPortal, Apelon Distributed Terminology System (DTS), Snow Owl, Ontoserver, Snowstorm, etc. Dans le monde francophone, le Laboratoire d'Enseignement et de Recherche sur le Traitement de l'Information Médicale (LERTIM) [14], Catalogue et Index des Sites Médicaux de langue Française (CiSMéF) [15] et la société informatique MONDECA<sup>1</sup> [16], tous trois spécialisés dans la représentation et le traitement des terminologies et des connaissances se sont associés pour concevoir et réaliser un « serveur multiterminologique de santé (SMT) » devenu présentement Health

---

<sup>1</sup> <https://mondeca.com/accueil/> (consulté le 21 juin 2021)

Terminology/Ontology Portal<sup>2</sup> (HeTOP). Chacun de ces serveurs comporte des caractéristiques, des qualités et des faiblesses propres.

La motivation principale du présent travail est de comparer les serveurs multiterminologiques (SMT) existants sur le marché afin de déterminer si l'un d'eux correspond aux besoins du Groupe de recherche interdisciplinaire en informatique de la santé (GRIIS). Ces besoins découlent d'un des projets du GRIIS, soit la Plateforme apprenante pour la recherche en santé et services sociaux (PARS3), un outil de partage délocalisé et sécuritaire des données qui permettrait la mise en place d'un SSA.

## **Objectifs**

Le GRIIS souhaite déterminer quel SMT offrirait les meilleures performances possibles dans le cadre d'une intégration à PARS3, en tenant compte des multiples utilisations qui pourront être faites de cette plateforme novatrice. Il faudrait par exemple faciliter l'utilisation de nombreuses terminologies hétérogènes, conserver l'exactitude sémantique lors de l'importation ou l'exportation de données à l'échelle nationale et internationale, pouvoir définir de nouvelles terminologies propres à des acteurs particuliers, etc.

À partir des besoins établis par le GRIIS, la présente étude compare l'utilisation et les fonctionnalités de serveurs existants, dans l'espoir de trouver un candidat qui conviendrait à PARS3. Si au bout de la comparaison, aucun SMT actuellement disponible ne satisfait les besoins du GRIIS, une évaluation déterminera s'il est plus avantageux pour le GRIIS de modifier un serveur existant ou d'en développer un nouveau sur mesure.

---

<sup>2</sup> <https://www.hetop.eu/hetop/> (consulté le 19 juin 2021)



## **Méthodologie**

Afin de répondre aux objectifs décrits ci-dessus, nous avons procédé de la façon suivante :

- ◇ Définition de la problématique du choix d'un SMT, sur la base d'un inventaire préliminaire et d'une recherche bibliographique.
- ◇ Recensement des besoins spécifiques d'un SMT à intégrer à PARS3 et dégagement des principaux cas d'utilisation, sur la base d'une revue de littérature et d'entrevues avec les utilisateurs/concepteurs de PARS3.
- ◇ Spécification des exigences et des fonctionnalités depuis les besoins recensés.
- ◇ Définition des critères de comparaison sur la base de protocoles définis.
- ◇ Inventaire et revue des principaux SMT existants.
- ◇ Étude et comparaison des SMT pour en dégager les modèles, les fonctionnalités et les règles de fonctionnement.
- ◇ Définition des fonctionnalités faisant défaut pour satisfaire les besoins définis.
- ◇ Évaluation du cout de développement d'un SMT par rapport au cout de modification du meilleur SMT candidat existant dont les sources sont disponibles.

## **Résultats**

Après avoir réalisé l'étude comparative, il est ressorti qu'aucun des trois serveurs étudiés ne répondait adéquatement à tous les besoins de GRIIS pour l'intégration avec PARS3. Cela a néanmoins permis de confirmer les fonctionnalités requises, tant celles retenues parmi celles proposées par l'un ou l'autre de SMT que celles faisant défaut.

À partir de ces résultats, il a fallu trancher à savoir s'il était plus profitable pour le GRIIS de développer son propre serveur ou d'améliorer un serveur existant à savoir LexEVS qui satisfait plus des critères par rapport aux autres serveurs, cela afin qu'il réponde aux exigences requises.

Finalement, nous sommes arrivés à des résultats qui démontrent que le développement d'un nouveau serveur serait plus profitable pour le GRIIS que de modifier LexEVS parce que cela serait moins cher à long terme.

## **Structure du mémoire**

Le présent document est structuré de la façon suivante :

- ◇ Le chapitre 1 présente une mise en contexte incluant les systèmes de santé apprenants, la problématique ciblée.
- ◇ Le chapitre 2 est consacré à la caractérisation des serveurs multiterminologiques ainsi qu'à la présentation d'un historique des systèmes terminologiques en médecine, à l'expression de besoins des utilisateurs de PARS3, à la définition des fonctionnalités et services ainsi qu'à l'étude des différents candidats disponibles retenus pour l'évaluation.
- ◇ Le chapitre 3 définit la méthode d'évaluation des critères de base pris en compte dans la comparaison ainsi que les protocoles à suivre.
- ◇ Le chapitre 4 explicite les résultats de la comparaison des serveurs multiterminologiques en se basant sur les critères définis au chapitre précédent.
- ◇ Le chapitre 5 présente une évaluation qui détermine s'il est plus avantageux pour le GRIIS de modifier un serveur existant ou d'en développer un nouveau sur mesure. Y sont présentés les modèles FP et Cocomo ainsi que le processus d'estimation ; le tout suivi de la conclusion de ce mémoire.

# Chapitre 1

## Présentation

Ce chapitre présente le contexte dans lequel ce mémoire s'inscrit, à commencer par le développement des SSA et leur rôle crucial dans le transfert de connaissances entre les activités de recherche et de soins. Cela exige toutefois une bonne capacité de partage de données, qui elle-même s'appuie sur les terminologies mises en correspondance par l'entremise des SMT. Nous présenterons alors les défis entraînés notamment par l'hétérogénéité des terminologies, desquels découlent les questions de recherche du présent mémoire.

### 1.1 Mise en contexte

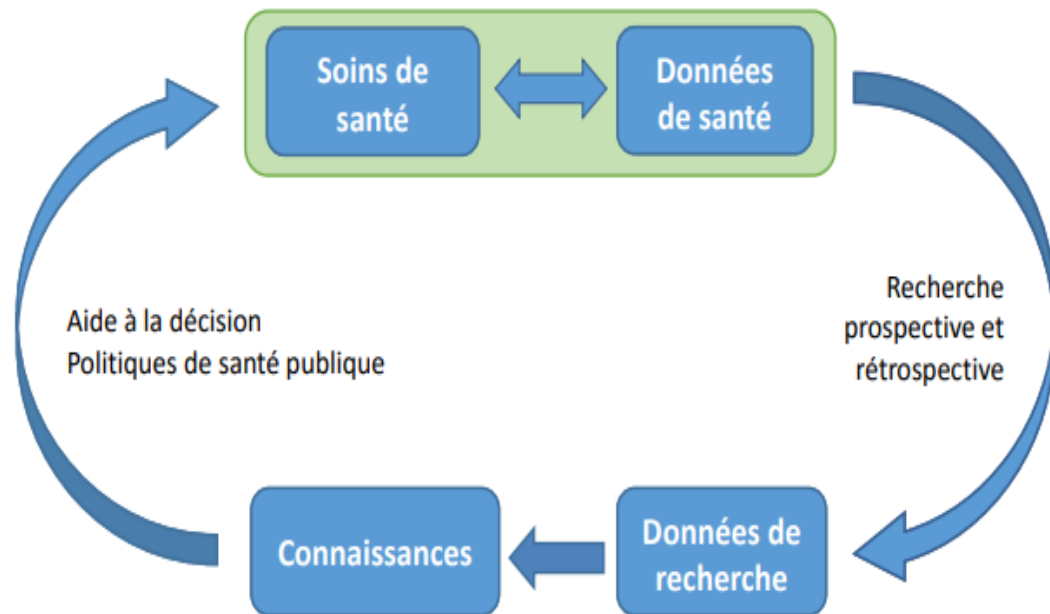
#### 1.1.1 Systèmes de santé apprenant

##### *Définition d'un SSA*

Un SSA a pour objectif de faciliter le transfert des connaissances entre les activités de recherche et les pratiques en santé afin d'améliorer les soins offerts à la population. Plus précisément, selon Ethier et coll. [17], un SSA se définit comme une vision d'un système de santé intégré dans lequel les progrès de la science, de l'informatique et de la culture des soins s'alignent pour générer de nouvelles connaissances en tant que produit de l'expérience de soins, en plus d'affiner et de mettre en œuvre de manière transparente les meilleures pratiques.

Pour y arriver, un SSA repose sur la collecte des données produites lors des soins, sur leur exploitation afin de produire de nouvelles connaissances, et sur l'exploitation de ces

connaissances dans le système de soins. Ces trois activités doivent être vues en termes de cycle [18] qui entraînent une amélioration continue de la santé et des soins.



**Figure 1 — Cycle d'un SSA [19]**

### ***Approches d'intégration des données au sein d'un SSA***

Il existe différentes approches pour connecter les systèmes sources aux systèmes utilisateurs [18].

#### *Entreposage de données*

Cette approche permet d'intégrer diverses sources de données dans un entrepôt de données commun. Cela se fait souvent à l'aide d'un processus ETL (Extract-Transform-Load, en français Extraction-Transformation-Chargement). Chaque jeu de données conserve toutefois la structure de leur source originale. Cela facilite l'analyse rétrospective en rassemblant des patients présentant des caractéristiques similaires dans des cohortes rétrospectives de patients. Cependant, au fur et à mesure que le

nombre de sources augmente, cette approche exige un niveau de généralisation identique qui va conduire soit à réduire considérablement le nombre de sources, soit à ignorer une quantité de données. De plus, des cadres règlementaires et juridiques distincts pour les institutions informatiques constituent un obstacle à la transmission de données.

#### *Fédération de données*

Cette approche permet le rassemblement des données de différentes sources tout en les structurant d'une manière identique. Les mêmes requêtes peuvent être effectuées sur chaque site et les données peuvent être agrégées facilement. Les données ne sont transmises que lorsqu'elles sont nécessaires et autorisées par les administrateurs des sources. Mais il est irréaliste d'exiger des institutions médicales qu'elles mettent elles-mêmes en place des structures de données et terminologies identiques, ou de changer l'ensemble de ses systèmes existants.

#### *Médiation de données*

Le principe de cette approche est d'éviter d'imposer une même structure de données et une même terminologie aux sources de données participantes pour sa mise en œuvre. Chaque source de données peut conserver sa propre structure et sa propre terminologie. Les données ne sont transmises que lorsqu'elles sont nécessaires et autorisées par l'administrateur des sources.

- **Global-as-view:** offre un modèle conceptuel central et une vue de l'union ou de la somme de chaque modèle local. Il est efficace, mais il y'a des risques d'asynchrone et d'incohérence dans un contexte où les sources ne sont prédéterminées.
- **Local-as-view:** offre un modèle conceptuel central, conçu indépendamment de toute source locale. Plus cohérent et stable que le global-as-view. Si une source locale est modifiée, seule la mise en correspondance entre cette source et le modèle central doit être mise à jour, tandis que les mises en correspondances des autres sources locales ne sont pas affectées. Néanmoins, la mise en

correspondance de chaque modèle de source de données sur un modèle central prend du temps et est plus complexe. Donc c'est un peu moins efficace que global-as-view.

Notons au passage qu'au niveau du modèle logique (souvent relationnel), la mise en œuvre de l'arrimage d'une source locale au modèle global consiste à définir (à tout le moins abstraitement) ce dernier en termes de « vues » sur le modèle local.

### ***PARS3 comme outil permettant la mise en place d'un SSA***

Développée dans le but de faire passer la recherche et les soins dans un contexte de SSA, la plateforme PARS3 est un ensemble de services coordonnés autour d'un même but : permettre un accès uniformisé et sécuritaire aux données de santé de patients, même lorsque ces données sont détenues par différents systèmes d'information clinique. L'objectif de PARS3 est de faciliter la réalisation des études cliniques tant rétrospectives que prospectives, d'aider la pratique réflexive et de permettre le développement d'outils d'aide à la décision. Pour ce faire, la plateforme offre un service d'accès aux données ontologiquement annotées. L'accès est réalisé selon un plan préalablement validé et approuvé. Afin de faciliter l'élaboration du plan, PARS3 doit interagir de façon privilégiée avec un serveur multiterminologique. Cela se fera par l'entremise d'un atelier, c'est-à-dire un ensemble d'applications et de logiciels. Cet atelier permet au responsable du projet d'en élaborer le plan (incluant notamment les critères de sélection de populations et les requêtes d'extraction) appelé « projet d'accès aux données » (PAD).

En d'autres mots, « PARS3 permet de traiter des données qui sont distribuées à travers des systèmes informatiques distincts et de les mettre en réseau de manière cohérente et compréhensible »<sup>3</sup>.

---

<sup>3</sup> GRIIS, « PARS3 • Solutions • GRIIS », GRIIS. <https://griis.ca/solutions/pars3/> (consulté le 05 mai 2021)

## 1.1.2 Terminologies

### *Définition*

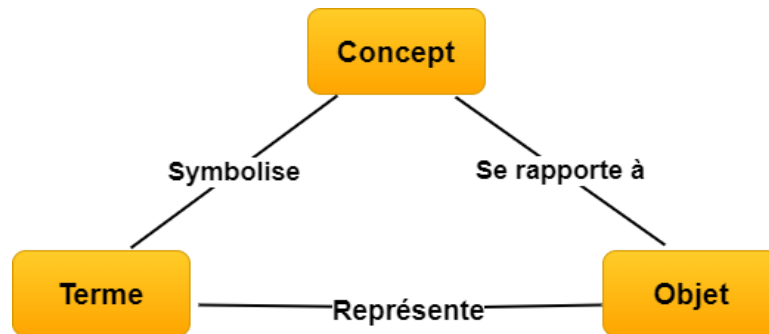
La norme ISO1087-1:2000 propose une définition adéquate et simple d'une terminologie [20], soit un « ensemble de désignations propre à une langue de spécialistes ». Les « désignations » peuvent être des termes de plusieurs types : synonymes, noms, symboles, etc.

De ce fait, le contenu et la structure d'une terminologie dépendent de la fonction pour laquelle cette terminologie va être utilisée. Dans une terminologie médicale (ou système terminologique médical), **des concepts** précis sont utilisés pour spécifier les concepts du domaine. **Des relations** peuvent exister aussi entre les concepts. Par exemple, des relations de généralisation-spécialisation sont prises en compte par plusieurs terminologies permettant de hiérarchiser les concepts du plus global au plus précis. Dans une terminologie, un concept peut être diversement « associé » à d'autres concepts. Nous parlons alors d'un « système de concept », qui est défini dans ISO1087:2019 (fr) [21] comme « un ensemble de concepts structuré dans un ou plusieurs domaines connexes selon les relations conceptuelles qui unissent ses concepts ». De plus, les terminologies peuvent être multilingues (toutes les formes équivalentes sous le même concept).

En d'autres mots, une terminologie recense **les concepts** d'un domaine précis (laboratoire, médicaments, etc.) et **des termes** qui les définissent [11]. Elle peut aussi représenter le résultat du recensement de concepts et termes. Les terminologies, en tant que résultat du recensement de concepts et termes, sont généralement de natures différentes. Cela est dû au fait que les objectifs de traitement de l'information médicale diffèrent d'un acteur à un autre ou encore d'un service à un autre. Ce qu'il faut retenir, c'est que toutes les terminologies sont construites autour du triangle sémiotique [11].

Dans un tel schéma, les objets sont des éléments du monde réel. Les objets peuvent être physiques ou abstraits. Ils sont regroupés ou catégorisés par des classes. Cette

catégorisation va faire référence à un ou des concepts. Ainsi, chaque concept va avoir une définition pour le clarifier. Cette définition est appelée « terme ».



**Figure 2 — Triangle sémiotique**

La construction d'une terminologie comprend principalement les aspects suivants :

- ◇ Analyser les concepts et les structures de concepts utilisés dans un domaine.
- ◇ Définir rigoureusement ces concepts.
- ◇ Faire l'inventaire des termes attribués aux concepts, y sélectionner un terme de référence.
- ◇ Établir des correspondances entre les termes dans les différentes langues, dans le cas d'une terminologie multilingue.
- ◇ Compiler la terminologie en construisant différents index pour en faciliter l'utilisation.

L'évolution d'une terminologie comprend principalement les aspects suivants :

- ◇ Créer de nouveaux concepts, de nouveaux termes, au besoin.
- ◇ Compléter les correspondances.
- ◇ Maintenir les index.
- ◇ Corriger des concepts existants, des termes existants au besoin.
- ◇ Corriger les correspondances en conséquence.
- ◇ Corriger les index en conséquence.
- ◇ Maintenir un inventaire des créations et des corrections.



### ***Des terminologies différentes pour des objectifs distincts***

Différents objectifs de traitement de l'information médicale ont amené à constituer des terminologies de natures différentes [22]. Pour citer quelques-unes :

- ◇ **Classification, recueil orienté de données** : représente un ensemble de termes organisés et hiérarchisés en classes et sous-classes [20]. La classification statistique internationale des maladies et des problèmes de santé connexes (**CIM** version 10), la classification commune des actes médicaux (**CCAM**) sont des exemples de classifications hiérarchiques médicales.
- ◇ **Thésaurus, recherche d'information** : il s'agit de terminologies dont les termes sont ordonnés et reliés entre eux par des relations hiérarchiques, d'équivalence ou d'association. Ce terme semble réservé aux terminologies documentaires : les termes, ou descripteurs, servent à indexer, ou décrire, des ressources documentaires [20]. La terminologie **MeSH** (Medical Subject Headings) est un thésaurus.
- ◇ **Nomenclature, recueil ouvert de données** : la notion de nomenclature est une autre variante de la notion générale de terminologie. Elle vise à recenser tous les concepts d'un domaine, sans se restreindre à priori à un objectif spécifique [23]. La nomenclature systématique des médecines humaine et vétérinaire (**SNOMED**) est une nomenclature multiaxiale qui permet de projeter les concepts médicaux selon plusieurs axes orthogonaux.

### ***La terminologie par opposition à l'ontologie***

Pour traiter une information médicale avec une « machine », il faut fournir un modèle formel [22]. Ce modèle est formé de l'ensemble des termes du langage et des relations qui permettent de relier des concepts généraux à des concepts plus précis. Plusieurs modèles existent, les principaux (pour le domaine médical) sont la terminologie et l'ontologie.

La comparaison entre terminologie et ontologie est un problème récurrent en ingénierie des connaissances et plus particulièrement en informatique médicale [23], car elles sont

toutes deux définies comme des ensembles de termes ou concepts reliés entre eux et spécifiques d'un domaine, et les deux définissent des propriétés aux concepts qui sont les attributs et les relations.

Les deux modèles se distinguent néanmoins, à commencer par le fait que, dans une terminologie, les définitions des termes sont généralement « descriptives » : elles transmettent comment les termes sont utilisés réellement dans certains domaines. En revanche, dans une ontologie, les définitions des termes sont « prescriptives » : elles transmettent comment les termes devraient être utilisés [24, p. 45]. Aussi, les ontologies sont structurées alors que les terminologies ne le sont pas toujours. Par ailleurs, les ontologies sont intrinsèquement dérivées d'un modèle plus ou moins formel, tandis que les terminologies n'ont pas forcément de modèle explicite, mais elles en suivent souvent un implicitement.

Ainsi, pour Diallo, « dans une terminologie, on s'intéresse aux termes et aux relations entre eux, la relation structurante de base et la relation d'hyponymie et son inverse l'hyponymie ; tandis que, dans une ontologie, on s'intéresse aux concepts et aux relations entre eux » [25, p. 15], où « hyperonymie » est une relation entre un terme général englobant des termes plus spécifiques.

Une telle approche aux terminologies est donc susceptible de poser des problèmes d'interopérabilité avec les ontologies. Plusieurs SMT proposent donc un modèle terminologique adapté : le concept y est identifié par un code unique et la définition lui est associée ; plusieurs termes pourront être associés au concept (par synonymie ou selon la langue) et, en général, un terme sera désigné comme terme de référence pour chaque langue. C'est à ce modèle terminologique que nous ferons désormais référence.

### **1.1.3 Serveurs multiterminologiques**

#### ***Définition***

Les SMT sont des ensembles d'outils permettant de définir, conserver, consulter et de mettre en correspondance plusieurs terminologies, leurs termes, leurs définitions et leurs codes. Ils permettent également de prendre en compte leur évolution et d'en répertorier les différentes versions.

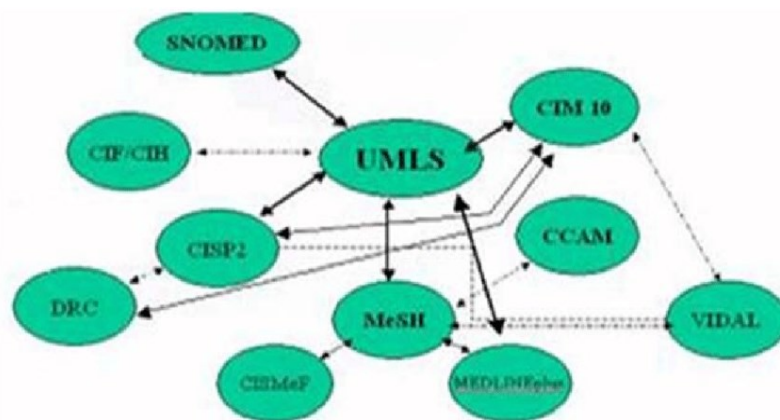
En pratique, les SMT doivent faciliter l'utilisation des terminologies pour un grand nombre d'applications, échanger des données avec d'autres serveurs, mettre en relation des concepts, fournir des méthodes de recensement des terminologies utilisées par sources et exécuter des requêtes diverses. En outre, les SMT doivent être capables d'exploiter les terminologies simultanément. Ces exploitations sont multiples et correspondent à des services particuliers en rapport avec des tâches utiles dans les SSA : aide au codage de l'activité des professionnels de santé, communication entre applicatifs utilisant des nomenclatures différentes, intégration dans des entrepôts de données de bases de données hétérogènes, etc. [13]. Parce qu'ils peuvent prendre en charge la gestion et la maintenance de plusieurs termes et l'organisation de ceux-ci dans plusieurs systèmes et organisations, leur adoption devrait se généraliser dans un avenir rapproché, autant en santé que dans d'autres domaines [26].

De plus, un SMT doit permettre d'emmagasiner différentes codifications avec les informations qui caractérisent les sources de données — par exemple, le nombre des patients, le nombre des médecins et quels sont les codes utilisés par ces sources.

#### ***Types des serveurs***

Il existe deux types de serveurs terminologiques :

- ◊ Les serveurs monoterminologiques (SMOT) qui hébergent une seule terminologie (exemple uniquement LOINC pour les résultats des examens de laboratoire ou uniquement CIM10 pour les maladies).
- ◊ Les serveurs multiterminologiques (SMT) qui hébergent non plus une seule terminologie à la fois, mais plusieurs. Ils sont très utiles pour le couplage de terminologie d'un même domaine. Avec cette mise en correspondance, les professionnels de santé utilisant des terminologies différentes pour le même domaine pourront s'échanger des données en toute transparence. Ces serveurs faciliteront aussi les alignements sémantiques [11] entre les différentes terminologies des différents domaines (voir Figure 3).



**Figure 3 — Relations existantes entre les terminologies médicales [11]**

Sur cette figure, les flèches représentent des relations et des liaisons entre les termes, grâce aux relations terminologiques, précisant la possibilité d'améliorer la recherche d'information et mieux répondre à la requête de l'utilisateur, et ce au moyen de la reformulation de la requête. Il est aussi possible de chercher toutes les liaisons possibles entre les termes (flèches avec des points) de la requête appartenant à une terminologie donnée et tous les termes des autres terminologies qui ont une correspondance avec les termes en question. Par exemple, à l'aide de la correspondance entre le terme MeSH « appareil correction auditive » et le terme SNOMED « prothèse auditive », il est possible d'enrichir le résultat et retrouver toutes les ressources indexées non seulement par le

terme « appareil correction auditive », mais aussi par le terme « prothèses auditives » [27].

À noter que, pour ce mémoire, nous nous intéressons seulement aux SMT, et non aux SMOT, qui ne correspondent pas du tout au besoin de mettre en correspondance plusieurs centaines, voire des milliers de terminologies, dans le cadre d'une utilisation avec PARS3.

## **1.2 Problématique**

### **1.2.1 Rôle du SMT en regard de PARS3**

Pour faciliter l'élaboration du plan d'accès aux données pour PARS3, ce dernier doit interagir avec un SMT par l'entremise d'un atelier PAD décrit précédemment. En outre, un SMT est un outil permettant aux utilisateurs de gérer et de stocker différentes terminologies.

L'atelier de rédaction de projet (ARP) appelé aussi atelier PAD voir Figure 4 permet au responsable de projet de rédiger un PAD syntaxiquement correct incluant les codes terminologiques appropriés en fonction des sources sélectionnées.

L'atelier de conseil éthique (ACE) est une application offrant au conseiller éthique la description et les spécifications d'un PAD. Aussi, il permet au conseiller de publier son avis éthique sur celui-ci pour approbation par le responsable d'accès aux données (RAD).

L'atelier de conseil clinique (ACC) est une application offrant au conseiller clinique la description et les spécifications d'un PAD. En outre, il permet au conseiller de publier son avis clinique sur celui-ci pour approbation par le RAD.

L'atelier de gestion des résultats (AGR) est une application offrant au chercheur ou à l'utilisateur des résultats un accès aux données publiées résultantes des requêtes PAD. Cet atelier pourrait aussi consulter le SMT pour vérifier la signification des codes reçus.

Ces quatre ateliers ont besoin de consulter un SMT afin de choisir, d'interpréter, de vérifier et de documenter les codes terminologiques utilisés par le PAD. Voir Figure 4 ci-après.

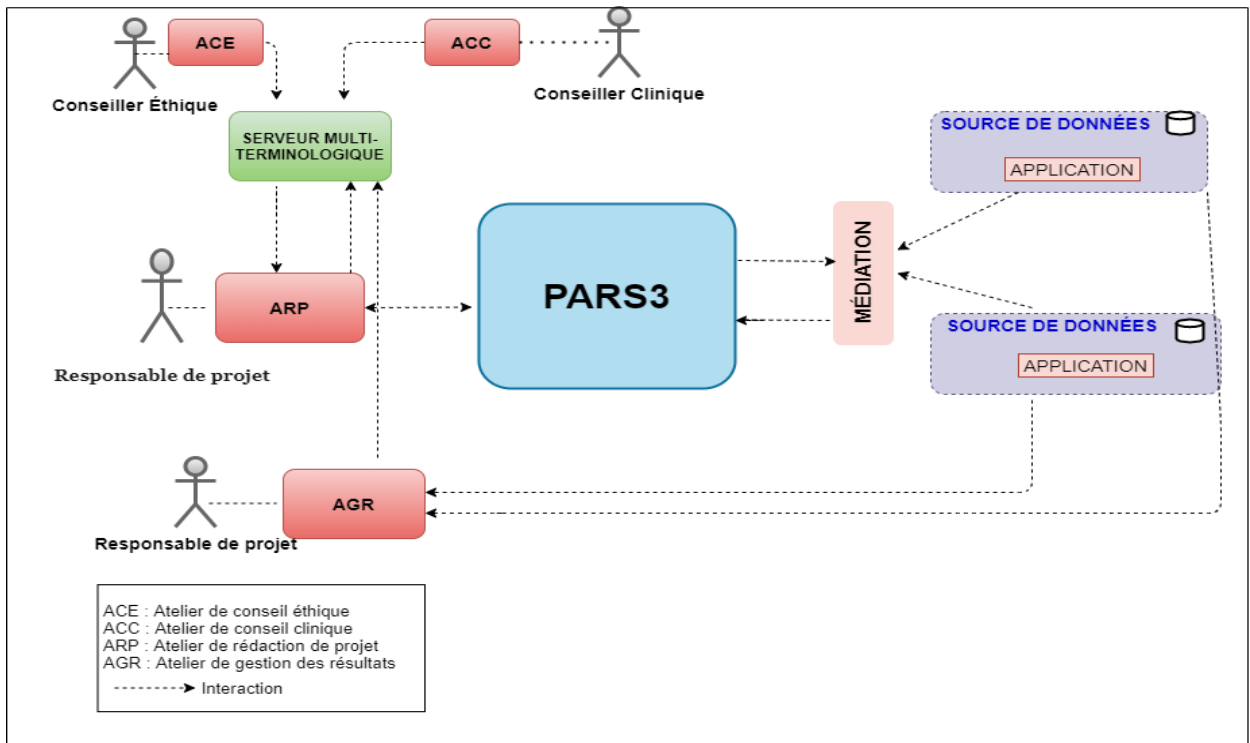


Figure 4 — Rôles du SMT en regard de PARS3

### 1.2.2 Caractéristiques du SMT recherché

Dans un contexte de SSA, les chercheurs et les autres utilisateurs potentiels désirent rassembler des données, ce qui est possible notamment grâce aux SMT. Ils sont cependant confrontés aux défis de l'hétérogénéité terminologique qui est la source de nombreux problèmes de mise en correspondance les sources des données et les ontologies. Du seul point de vue terminologique, d'une part, il existe plusieurs standards terminologiques concurrents, d'autre part, plusieurs sources utilisent leur propre terminologie.

C'est par exemple le cas de diabète, théorisé de plusieurs façons différentes depuis 1950, et où chaque théorie a eu sa propre classification et chacune des classifications, plusieurs

codifications. La codification utilisée (voire les codifications utilisées) dans un dossier de patient dépendra alors du moment de sa création, de celui des mises à jour qui ont été faites et des systèmes utilisés. Ainsi, le chercheur doit pouvoir connaître les codifications utilisées par les sources qu'il entend consulter (et la terminologie et la théorie sous-jacente qui leur sont associées) afin d'établir une requête de sélection de patients diabétiques répondant aux critères de l'étude qu'il entreprend. Pour chaque terminologie, il faut donc une classification claire, qui explique les grandes familles de codes et ce à quoi ils correspondent.

En conclusion, un système d'accès aux données (comme PARS3) doit faire en sorte que les données cliniques puissent être documentées de telle sorte que, malgré leur hétérogénéité, elles puissent être rendues interopérables [28].

Le SMT est la composante qui doit résoudre ou pallier les problèmes suivants :

- ◇ Stockage des terminologies — Le SMT doit permettre de stocker et de documenter différentes terminologies ainsi que de les rendre disponibles à la demande des utilisateurs.
- ◇ Stockage des mises en correspondance — Les chercheurs n'ont pas la même vision de recherche, ce qui pose un problème d'automatisation de mise en correspondance des termes de terminologies différentes, d'une part parce que ce ne sont généralement pas des bijections, et d'autre part parce que la sélectivité et la spécificité dépendent du type d'étude désiré. Il n'en demeure pas moins utile de pouvoir stocker une correspondance entre les termes d'une même terminologie ou entre ceux de terminologies différentes, que cette correspondance soit partielle, inclusive ou équivalente.
- ◇ Multilinguisme — Les concepts ne sont pas toujours dans la langue de l'utilisateur. Il faut donc les traduire, ce qui peut causer des erreurs sur les libellés et la sémantique. La conséquence première est la perte de concept [11]. Un support adéquat du multilinguisme et du processus de traduction est donc requis.

- ◊ Variation terminologique — Malgré la disponibilité de terminologies cliniques normalisées (dont SNOMED CT), de nombreux systèmes d'information clinique utilisent des codes personnalisés, supplémentaires, voire modifiés [12]. Il est donc souhaitable de pouvoir facilement définir une terminologie particulière comme une variation d'une autre.

Par ailleurs, le SMT devra pouvoir faire face à un contexte d'utilisation exigeant :

- ◊ Volume des données terminologiques — Certaines terminologies incluent une grande quantité de contenu, ce qui rend difficile la recherche d'un concept précis puisque cela affecte la performance.
- ◊ Fréquence des mises à jour des terminologies — L'accès au contenu d'une terminologie implique généralement un processus d'indexation qui peut être long et coûteux. La fréquence des mises à jour et l'apparition de nouvelles versions imposent en général une réindexation des contenus, d'où l'importance d'optimiser ce processus afin de garantir une bonne disponibilité du service [12].

### **1.3 Question de recherche**

L'objectif global du présent mémoire est de déterminer quel SMT répondrait le plus adéquatement au problème des terminologies hétérogènes dans le contexte de la plateforme PARS3, en tenant compte de cas d'utilisation ciblés.

Ainsi, dans un premier temps, nous désirons comparer les SMT actuels afin de déterminer s'il existe déjà un candidat répondant à ses besoins, qui sont :

- ◊ la possibilité d'utiliser des terminologies externes existantes ou à venir et de les mettre à jour ;
- ◊ la possibilité de créer et mettre à jour des terminologies propres ou d'ajouter des termes ;



- ◇ la capacité de stocker des relations entre terminologies et entre codes de terminologies différentes ;
- ◇ la possibilité de rechercher avec différents critères ;
- ◇ la possibilité d'utiliser plusieurs terminologies différentes à la fois ;
- ◇ la possibilité de recenser des catalogues des terminologies par source de données ;
- ◇ la prise en charge multilingue, car le serveur doit permettre de décrire les concepts dans les langues souhaitées (pour le moment, au moins le français et l'anglais) ;
- ◇ la capacité d'accéder au SMT et à ses fonctionnalités autant au moyen d'une interface programmatique (API) que d'une interface personne-machine (IPM) ;
- ◇ la sécurité du serveur (authentification et contrôle d'accès) ;
- ◇ la prise en charge des mises à jour externes publiées par des organisations externes responsables de terminologies standardisées ou normalisées (comme SNOMED CT, qui est mise à jour deux fois par an).

La réponse à ces questions est en partie fournie par une ou plusieurs métriques qui indiquent l'adéquation de chaque serveur par rapport à un ensemble de critères.

Dans un deuxième temps, si la comparaison révèle que tous ces besoins ne peuvent être comblés par un SMT existant, le mémoire évaluera si la meilleure façon de répondre aux besoins de PARS3 passe par l'adaptation du SMT le plus satisfaisant selon la comparaison, ou s'il faudrait plutôt envisager le développement d'un nouveau SMT.

## Chapitre 2

### Les serveurs multiterminologiques dans les SSA

Ce chapitre s'intéresse plus en détail à l'objet d'étude de ce mémoire, c'est-à-dire aux serveurs multiterminologiques. Nous survolerons ainsi leur historique ainsi que leur caractérisation plus générale, puis nous présenterons les serveurs actuellement disponibles qui ont été considérés pour le présent projet de recherche.

#### 2.1 Historique des systèmes terminologiques en médecine

On trouve des traces de classifications systématiques dès le IV<sup>e</sup> siècle avant Jésus Christ, avec Aristote notamment, mais il faudra attendre le début du XVII<sup>e</sup> siècle pour qu'un début de terminologie médicale voie le jour, dans le cadre « **du London Bills of mortality** ». Dans cet hebdomadaire, les décès sont recensés, selon leurs causes. En 1771, à Paris, paraît une nosologie méthodique qui constitue la première classification aboutie des maladies [29]. Reprise par Pinel (1798-1802), puis par Bertillon (1893), elle est finalement l'ancêtre des classifications internationales des maladies et des problèmes de santé connexes. Ces classifications sont principalement utilisées à des fins statistiques et épidémiologiques, elles permettent de s'assurer que les cas sont bien définis de la même manière, quel que soit le médecin établissant le diagnostic [29].

En 1929, pendant le symposium de l'académie de médecine de New York, une nomenclature standardisée des maladies est conçue. La nouveauté de ce système de représentation réside dans l'existence de plusieurs axes : pour décrire une maladie, le médecin utilise à la fois un code de topologie/anatomie et un code d'étiologie/physiopathologie. C'est le début de la compositionnalité, où toutes les maladies n'ont plus à être énumérées et prévues. À noter que ce principe sera conservé

des décennies plus tard, lors du développement de la « *Systematized NOmenclature of MEDicine* » (SNOMED), dont la version 3.5 est organisée en hiérarchie selon 11 axes, avec plus de 100 000 concepts, couvrant les champs de la médecine, de la dentisterie humaine, ainsi que de la médecine vétérinaire.

Avec l'apparition de l'informatique, l'utilisation de systèmes terminologiques devient une évidence. Cependant, pour que l'ensemble des connaissances contenues dans ces systèmes terminologiques soit exploitable par des machines, il est nécessaire de formaliser les définitions des concepts et les relations qui les lient les uns par rapport aux autres. Ainsi — et c'est le stade ultime de développement actuel — les systèmes terminologiques progressent par le biais de relations riches et formelles.

## **2.2 Caractérisation des serveurs multiterminologiques**

### **2.2.1 Rôle et missions**

La mise en place d'un serveur multiterminologique est prérequis pour la gestion des différentes ressources sémantiques [30]. Devant la multiplicité des terminologies disponibles, il apparaît nécessaire de mettre en place une solution rendant celles-ci « interopérables ». Cela implique le développement d'un référentiel sémantique commun permettant l'interaction efficace des différentes terminologies, au sein d'un système capable de les gérer toutes simultanément. Seul un SMT s'avère être assez performant pour remplir ces conditions [30]. Les SMT sont utilisés pour acquérir de la connaissance, pour faciliter la prise de décision et la recherche.

Selon les travaux menés par le Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en e-Santé (LIMICS<sup>4</sup>), le SMT pourrait se présenter sous la forme d'un logiciel, associé à un espace d'archivage numérique, permettant le stockage, la gestion et

---

<sup>4</sup> <http://www.limics.fr/> (consulté le 11 février 2021)

la diffusion de plusieurs terminologies. Il s'agirait alors d'un type de serveur plus puissant, mais aussi plus complexe à utiliser.

Quatre missions principales attribuables à un SMT sont identifiées dans le rapport de l'Agence française de la santé numérique [30] :

- ◊ Le stockage de données en vue d'une diffusion uniformisée ;
- ◊ L'administration des différentes ressources terminologiques ;
- ◊ La gestion et la maintenance globales des terminologies ;
- ◊ L'utilisation des terminologies à des fins industrielles ou de recherche.

Un SMT comporte alors deux dimensions :

1. La gestion de la production de contenus (masquée pour l'utilisateur) : réception des ressources sémantiques, gestion de leur interopérabilité et administration des mises à jour, validation et contrôle avant diffusion ;
2. La diffusion des ressources sémantiques : diffusion auprès des utilisateurs des contenus et des mises à jour.

### **2.2.2 Méthodes d'accès typiques**

Pour fournir des services à des applications et à des humains, un SMT va se baser sur les standards, normes et formats existants, par exemple en publiant les informations à travers les services web de type Simple Object Access Protocol (SOAP) ou Representational State Transfer (REST), avec un format de données structurées dans un fichier eXtensible Markup Language (XML), JavaScript Object Notation (Json) ou autre.

Plusieurs interfaces de consultation des terminologies sur un SMT existent. Elles sont divisées en deux catégories, soit :

- ◊ **L'Interface personne-machine (IPM)** : Elle a pour vocation la navigation réalisée par une personne utilisatrice entre les différents concepts, la gestion des

terminologies et leurs contenus ainsi que la gestion du SMT, incluant la création, la modification, la navigation et la suppression de terminologies.

- ◊ **L'interface de programmation d'application (ou interface programmatique, généralement désignée par le sigle API, pour *Application Programming Interface*)**. Son rôle est d'exposer les fonctions qui permettent de définir, manipuler et gérer les terminologies. L'API peut être utilisée par les applications pour récupérer des informations sur des concepts. Les API incluent entre autres, la gestion de niveau de base de concepts d'une terminologie, tels que l'interrogation, la suppression ou même l'ajout de nouveaux concepts. En outre, des outils administratifs tels que l'importation et l'exportation de données sont fournis. La granularité des API varie, car toutes ne suivent pas une norme standard et sont donc laissées aux développeurs pour les implémenter comme ils le souhaitent. Dans le cadre de l'utilisation d'un SMT en santé, les API travaillent avec des systèmes d'information basés sur des classifications nationales bien établies dans la pratique clinique.

### ***API REST***

REST (*Representational State Transfer*) est un modèle standardisé d'API très fréquemment utilisée par les SMT. Pour cette raison nous la présentons ici.

REST est définie par « un ensemble de standards et de lignes directrices architecturales qui structurent la façon de communiquer les données entre une application et le reste du monde, ou entre les différents composants d'une application »<sup>5</sup>. Un système logiciel (une application, un service, voire une bibliothèque) est dit RESTful lorsqu'il respecte les principes du modèle REST et en particulier que la partie serveur et la partie client

---

<sup>5</sup> <https://openclassrooms.com/fr/courses/6573181-adoptez-les-api-rest-pour-vos-projets-web/6817216-identifiez-les-avantages-d-une-api-rest> (consulté le 10 avril 2021)

communiquent sans que le client ne connaisse la structure et le contenu des informations stockées sur le serveur.

Il est basé sur les technologies de base du Word Wide Web, principalement le protocole HTTP (*HyperText Transfer Protocol*) avec l'URI (*Uniform Resource Identifier*) et les types MIME (*Multipurpose Internet Mail Extensions*).

Parmi les avantages clés des API REST, on peut citer les suivants :

- ◊ la minimisation du couplage entre le client et le serveur ;
- ◊ le fait que la communication ne repose pas sur un état partagé (*stateless*), ce qui rend les requêtes API très spécifiques et orientées vers le détail ;
- ◊ la possibilité pour le client de mettre des données en cache, et donc de ne pas devoir constamment faire des requêtes aux serveurs<sup>6</sup>.

Toutes les API documentées sur les serveurs terminologiques comparés dans ce mémoire utilisent REST dans leur implémentation.

Lorsqu'une requête est effectuée via une API RESTful, elle transfère une représentation de l'état de la ressource au demandeur. Ces informations, ou représentation, sont fournies dans l'un des nombreux formats via http : JSON (JavaScript Object Notation), HyperText Markup Language (HTML), ou autre texte brut. Les utilisateurs ne sont pas obligés de connaître les paramètres de requête dans un ordre spécifique.

Les API REST permettent une exécution simple des appels de fonctions, avec un environnement de test intégré dans lequel l'utilisateur peut essayer d'exécuter des requêtes et d'en comparer les résultats de manière simple, directement dans le navigateur. Dans le cas des serveurs multiterminologiques, il arrive fréquemment

---

<sup>6</sup> Ceci est également un inconvénient majeur en regard de la mise à jour des données. Pour cette raison, l'architecture REST n'est pertinente en pratique que dans un contexte de consultation ou de modification atomique entièrement validable par le serveur.

qu'aucune documentation concernant les appels d'API n'existe chez le fournisseur et qu'elle se trouve plutôt dans l'appel d'API REST lui-même, sous la forme d'exemples de requêtes et de réponses [1].

### **2.2.3 Fonctionnalités usuelles**

À partir des missions citées dans la section 2.2, nous posons que les fonctionnalités usuelles pour un SMT sont la création, la conservation, la consultation, la modification, la navigation et la mise en correspondance des terminologies. En dehors de cela, un SMT peut aussi fournir d'autres fonctionnalités, telles que la prise en charge de la publication et de la récupération des terminologies. Toutes ces fonctions doivent être accessibles autant via une IPM qu'une API.

Nous énonçons qu'il existe deux approches principales pour implémenter les fonctionnalités d'un SMT (inspiré de [31]) :

- ◇ Première approche : les outils intégrés.
- ◇ Deuxième approche : les API.

La première approche consiste à développer tout type d'application au-dessus du serveur terminologique, comme un navigateur de terminologie, un éditeur de terminologie, un traducteur de terminologie, etc. Dans cette approche, ces outils exécutent généralement des aspects des fonctionnalités du SMT. Par exemple, *National Cancer Institute* (NCI) a développé un navigateur de terminologie pour permettre aux utilisateurs de parcourir et consulter les différentes terminologies stockées dans son serveur multiterminologique. De plus, de chercher un terme avec sa définition, ses synonymes et ses relations avec d'autres termes, etc.

La seconde approche, celle recourant aux API, peut être utilisée pour écrire un programme qui fournit certains services au SMT. En règle générale, toutes les applications ou tous les outils peuvent utiliser ces API pour interfacer le SMT. Par

exemple, NCI a fourni quelques API pour accéder aux terminologies stockées dans son serveur multiterminologique.

## 2.2.4 Principaux composants et architectures typiques

Un SMT peut être présenté sous la forme d'une application web qui comprend souvent les composants suivants :

- ◇ Un outil fournissant l'accès à la base de connaissances terminologiques résidant dans une base de données. Les fonctionnalités de création, conservation, consultation, modification, navigation et mise en correspondance sont fournies par cet outil.
- ◇ Un navigateur Web pour afficher le contenu terminologique. Ceci est une IPM permettant de visualiser le contenu dans la base de données.

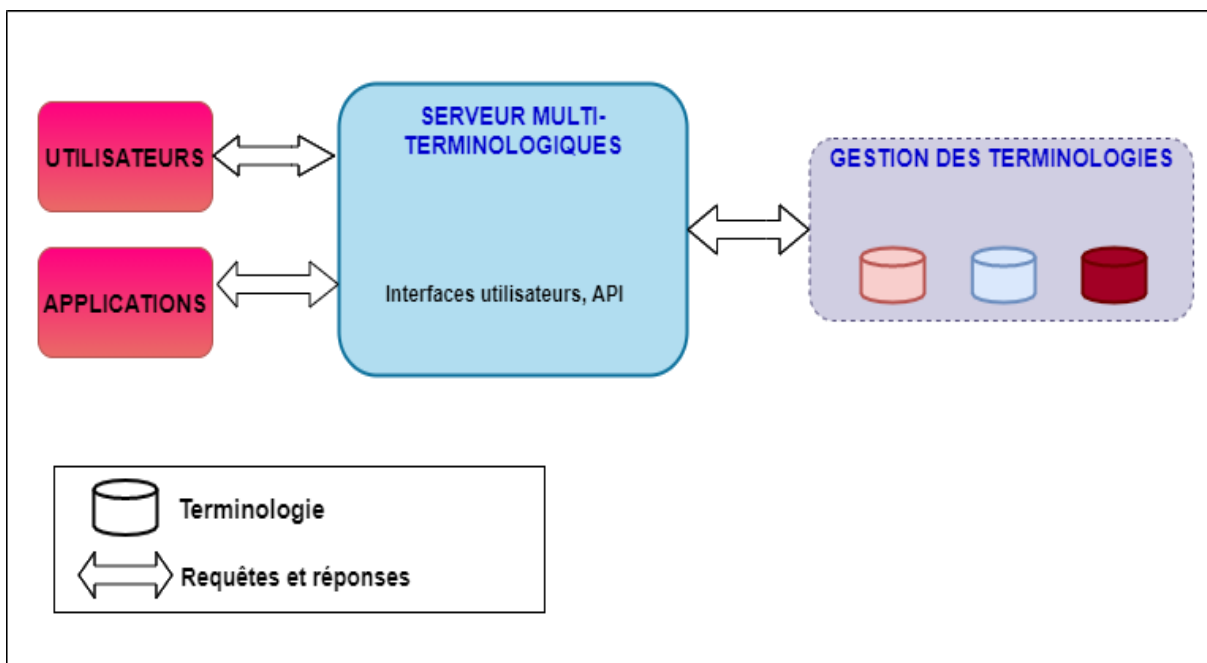


Figure 5 — Architecture et services du SMT



## **2.3 Expression des besoins**

La recherche des besoins est fondée sur la détermination des besoins directement associés à PARS3 et découlant des cas d'utilisation identifiés dans un contexte d'accès aux données au sein d'un SSA. Les besoins n'ont volontairement pas été distingués sur la base des interfaces (IPM ou API) susceptibles de les porter afin de permettre un développement indépendant de toutes les applications relatives aux terminologies, ceci en conformité avec l'approche préconisée par le GRIIS.

### **2.3.1 Besoins généraux**

Dans le cadre de PARS3, le SMT doit pouvoir :

- ◇ faciliter la création, la publication, la recherche, l'accès, la mise en correspondance et la traçabilité des terminologies ;
- ◇ rendre les terminologies interopérables (donc, être en mesure de les exprimer selon un modèle unique) ;
- ◇ faciliter la maintenance et la publication de nouvelles versions des terminologies ;
- ◇ faciliter la saisie et le maintien des terminologies, concepts, termes, définitions et codes de sources spécifiques.

### **2.3.2 Besoins exprimés par les utilisateurs**

Afin de mieux comprendre les besoins, nous nous sommes entretenus avec des personnes susceptibles d'être les principaux utilisateurs de l'outil. Ceux-ci nous ont expliqué leurs attentes concernant le SMT sous forme de cas d'utilisation. Plusieurs des fonctionnalités requises par ces cas d'utilisation sont déjà mises à disposition par certains SMT, attendu une adaptation pour qu'ils puissent interagir avec PARS3. Toutefois, plusieurs améliorations, voire quelques carences, ont également été soulignées.

Les sections suivantes prennent compte des besoins ainsi dégagés des cas d'utilisation proposés par les utilisateurs. En regard du SMT, ces besoins seront explicités à l'aide d'exigences et de critères d'évaluation au Chapitre 3.

### ***Utilisateurs ciblés***

- ◇ **Chercheur** : Personne qui consulte les terminologies hébergées dans le serveur pour nourrir sa recherche (au titre de responsable de projet ou d'utilisateur des résultats, par exemple).
- ◇ **Clinicien** : Personne qui consulte les terminologies pour nourrir sa pratique de soins (au titre d'utilisateur des résultats, par exemple).
- ◇ **Conseiller** : Personne qui contribue aux tâches des autres acteurs, notamment pour vérifier les PAD (au titre de conseiller éthique ou de conseiller clinique, par exemple).
- ◇ **Documentaliste** : Personne qui participe à la collecte ou à la saisie d'information médicale et qui alimente les sources exploitées par PARS3.
- ◇ **Gestionnaire** : Administrateur ou superviseur du SMT qui donne le droit d'accès aux autres utilisateurs et qui s'occupe de la gestion des terminologies.
- ◇ **Ontologiste** : Utilisateur du SMT responsable de la structure des terminologies et des définitions formelles des concepts.
- ◇ **Terminologue** : Utilisateur du SMT qui crée les définitions linguistiques des concepts, la mise en correspondance entre ceux-ci, etc.

### ***Besoins du SMT exprimés par les utilisateurs***

Les besoins se dégageant des cas d'utilisation présentés par les utilisateurs sont résumés dans le Tableau 1 suivant :

**Tableau 1 — Synthèse des besoins exprimés par les utilisateurs**

Catégorie	Besoins	Utilisateurs
Gestion	Pouvoir gérer les terminologies : création, suppression et lecture des terminologies.	Gestionnaire, Terminologue
	Pouvoir gérer les concepts : création, mise à jour, suppression et lecture des concepts.	
	Pouvoir gérer et mettre à jour des terminologies sur la base de sources externes.	Gestionnaire
	Pouvoir définir les étiquettes des concepts : définir le code du concept, ajouter un concept, supprimer un concept, etc.	Terminologue
	Pouvoir traduire et gérer la traduction des termes et définitions et étiquettes.	
Sélection	Lors de la rédaction du PAD, pouvoir déterminer les termes et les codes adéquats en fonction de la variété terminologique des sources.	Chercheur
Consultation	Obtenir le catalogue des terminologies	Terminologue, Ontologiste, Chercheur, Clinicien, Documentaliste, Conseiller
	Rechercher au sein d'une terminologie	
	Obtenir les métadonnées d'une terminologie	
	Consulter les différences entre deux d'une terminologie	
	Obtenir une liste des concepts d'une terminologie	
Mise en correspondance	Rechercher au sein d'un groupe de terminologies	Terminologue, Ontologiste, Chercheur
	Afficher un concept	
	Afficher la structure hiérarchique d'une terminologie et pouvoir y naviguer	
	Pouvoir mettre en correspondance les concepts d'une terminologie (intraterminologique)	
Mise en correspondance	Pouvoir mettre en correspondance les concepts des terminologies différentes (interterminologique)	Terminologue, Ontologiste
	Pouvoir mettre en correspondance les sources, les ontologies et les terminologies (recensement des terminologies par source).	Terminologue, Ontologiste, Chercheur
	Pouvoir importer des terminologies dans différents formats standards OWL, XML, etc.	Gestionnaire
Stockage	Pouvoir exporter des terminologies dans différents formats standards OWL, XML, etc.	Gestionnaire, Terminologue, Ontologiste, Chercheur

## 2.4 Fonctionnalités et services

À partir des besoins exprimés, nous avons établi cinq types de ressources principales (système de codification, jeu de valeur, alignement, contexte d'utilisation, concept) et six groupes d'activités en permettant l'exploitation (GT, GC, GV, GAT, GAC, IE). Nous proposons ensuite une division opérationnelle des SMT en trois services (stockage, consultation, gestion) regroupant l'ensemble des fonctionnalités requises à la réalisation des activités composant les groupes retenus. Une telle division opérationnelle a pour but de faciliter la comparaison des SMT, car ils sont le plus souvent présentés (et documentés) selon une structure opératoire que fonctionnelle.

### 2.4.1 Définitions des types de ressources principales

Les ressources principales dérivées de l'analyse des besoins sont :

- ◇ **Système de codification (SC)** : est une structure générique qui contient des métadonnées terminologiques, telles que le nom, la description et les versions. Les terminologies sont représentées par un système de codification. Un système de codification est aussi connu comme système de codes, ontologie, terminologie, etc.
- ◇ **Jeu de valeurs (JV)** : aussi appelé ensemble de valeurs, spécifie un ensemble de codes tirés d'un ou plusieurs systèmes de codification, destiné à être utilisé dans un contexte particulier. Les jeux des valeurs peuvent être sélectionnés à partir de différents systèmes de codification. Dans certains cas, un jeu de valeurs unique peut également inclure des concepts de différents systèmes de codification.
- ◇ **Alignement** : consiste à mettre en correspondance les concepts d'un système à un autre. Un alignement est une connexion qui relie les concepts entre eux dans des graphiques représentant des informations terminologiques/ontologiques. Les alignements sont construits comme une combinaison de trois concepts : un concept source, un concept de type relation (association) et un concept cible.
- ◇ **Contexte d'utilisation (CU)** : permet de spécifier et de déterminer la liste de concepts ou la description locale (d'un concept spécifique) à utiliser dans un

contexte d'application donné. Les contextes d'utilisation sont un attribut de listes de concepts et des descriptions locales.

- ◇ **Concept** : est une représentation unique d'une idée dans une terminologie représentée par un système de codification.

### 2.4.2 Définition des groupes d'activités principales

Les groupes d'activités principales mettant en cause les ressources principales dérivées de l'analyse des besoins sont :

- ◇ **GT** : Prise en charge de la consultation, de la création, de la modification et de la suppression des terminologies via une IPM ou une API.
- ◇ **GC** : Prise en charge de la consultation, de la création, de la modification et de la suppression des contenus des terminologies (concepts, description, synonyme, etc.) via une IPM ou une API.
- ◇ **GV** : Maintien et stockage de plusieurs versions (y compris non publiées et publiées) pour chaque terminologie et fournir des API pour y accéder tous.
- ◇ **GAT** : Prise en charge de la création, de la modification, de la consultation et de la suppression des alignements (mise en correspondance) entre des terminologies via une IPM ou une API.
- ◇ **GAC** : Prise en charge de la création, de la modification, de la consultation et de la suppression des alignements entre des concepts des terminologies via une IPM ou une API.
- ◇ **IE** : Prise en charge d'importation et d'exportation des terminologies et leurs contenus par des API ou IPM, etc.

### 2.4.3 Définition des services

Afin de mener l'évaluation à bien, les fonctionnalités des SMT seront examinées selon trois services principaux : stockage, consultation et gestion. Chaque service comporte des fonctionnalités propres.

### ***Service de stockage***

Il s'agit d'un ensemble de fonctionnalités permettant de gérer le contenu dans le cadre d'un service de terminologie. Les fonctions de stockage incluent la possibilité de charger des terminologies, d'exporter des terminologies, d'activer des terminologies et de retirer des terminologies. Ces fonctions sont généralement protégées et accessibles par les administrateurs de services disposant des autorisations appropriées.

Ce service comprend deux fonctionnalités principales décrites dans la section suivante :

- ◇ **Fonctionnalités d'importation** : doit permettre de charger ou d'importer la mise à jour complète ou incrémentielle des différents éléments, soit les systèmes de codification, des jeux de valeurs, et des alignements. Il doit offrir également la possibilité d'activer et de désactiver le contenu chargé.
- ◇ **Fonctionnalités d'exportation** : doit permettre d'exporter le contenu complet ou partiel du système de codification, des jeux de valeurs ou des alignements.

### ***Service de consultation***

Il s'agit d'un ensemble de fonctionnalités qui permet de rechercher du contenu en fonction de certains critères de recherche. Cela inclut des restrictions à des alignements spécifiques ou à d'autres attributs de la terminologie, y compris la recherche des alignements pour les jeux de valeurs. Cela représente l'utilitaire principal pour l'utilisation du contenu terminologique dans un certain nombre de contextes d'application.

Le service de consultation permet d'interroger le contenu disponible dans les services de terminologies. Il comprend des fonctionnalités principales décrites dans les sections suivantes :

- ◇ **Fonctionnalités de consultation d'alignement** : offrir la possibilité d'interroger les alignements disponibles dans le système et obtenir les détails sur ces alignements.

- ◊ **Fonctionnalités de consultation de système de codification** : offrir la possibilité d'interroger le contenu des terminologies qui sont disponibles dans le serveur, de lister tout le contenu et obtenir des détails sur le contenu.
- ◊ **Fonctionnalités de consultation de contexte d'utilisation** : offrir la possibilité d'interroger le contexte d'utilisation disponible dans le système et également d'interroger la liaison entre le jeu de valeurs et le contexte d'utilisation.
- ◊ **Fonctionnalités de consultation de jeu de valeurs** : offrir la possibilité d'interroger les jeux de valeurs disponibles dans le système.

### ***Service de gestion***

Il s'agit d'un ensemble de fonctionnalités qui permet de créer et de gérer du contenu. Du point de vue du service de terminologie, cela inclut les fonctionnalités appropriées pour créer, modifier ou supprimer des concepts, des alignements, etc. Cela inclurait également le traitement des événements de changement de divers fournisseurs de terminologie.

Le service de gestion comprend des fonctionnalités plus particulièrement applicables aux concepts et aux jeux de valeurs :

- ◊ **Créer** — Possibilité de créer une nouvelle entrée. Par exemple, ajouter un nouveau concept à un système de codification, ajouter une nouvelle propriété à un concept, etc.
- ◊ **Modifier** — Possibilité de modifier les attributs d'une entrée existante. Après chaque modification, la version de l'entrée modifiée sera modifiée. L'état d'entrée de chaque version sera enregistré.
- ◊ **Supprimer** — Possibilité de supprimer une entrée existante. Cette opération supprimera définitivement l'entrée du système.
- ◊ **Changer le statut** — Le statut, la date d'entrée en vigueur, la date d'expiration, etc. sont les attributs de version de concept. Cette opération permet de modifier les attributs de version d'un concept. Exemple : le statut d'un concept peut être changé en « Retraité », où on peut désactiver un concept, etc.

Le service de gestion comprend également des fonctionnalités plus spécifiquement applicables aux systèmes de codification, aux alignements et aux contextes :

- ◇ **Fonctionnalités de gestion de système de codification** : offrir la possibilité de créer, modifier et supprimer des systèmes de codification et leur contenu.
- ◇ **Fonctionnalités de gestion d'alignement** : offrir la possibilité de créer, modifier et supprimer les alignements entre les entités de systèmes de codification.
- ◇ **Fonctionnalités de gestion de contexte d'utilisation** : offrir la possibilité de créer, modifier et supprimer des contextes d'utilisation et de les intégrer à un système de codification.

## 2.5 Les SMT considérés dans ce mémoire

Sur la base des serveurs disponibles, SNOMED International a dressé en 2021 la liste<sup>7</sup> des principaux SMT permettant l'exploitation de SNOMED CT. Ces résultats ont été complétés en y incluant d'autres SMT, présentement disponibles, tels que LexEVS, BioPortal, HeTOP, etc.

Plusieurs SMT ont été éliminés d'office parce qu'il était évident dès le départ qu'ils ne pourraient pas satisfaire les besoins retenus. C'est le cas, par exemple, de Snowstorm et West Coast, qui sont des SMOT, et qui n'auraient par conséquent pas pu servir avec plusieurs terminologies comme le requiert PARS3. D'autres, tels que Terminz (NZ exemplar) et ITServer n'offraient pas une documentation suffisante et n'auraient par conséquent pas pu être déployés pour être comparés.

Il en résulte une liste de cinq SMT qui sont présentés ci-après, soit LexEVS, HeTOP, Snow Owl, Ontoserver et BioPortal.

---

<sup>7</sup> <https://confluence.ihtsdotools.org/display/FHIR/Features+of+Known+Servers> (consulté le 30 novembre 2020)



## 2.5.1 SMT intéressants

### *HeTOP*

HeTOP [32] est un serveur multiterminologique conçu pour effectuer des recherches terminologiques. Ce serveur n'est pas en code ouvert, mais il est mis à la disposition des utilisateurs gratuitement tant que son utilisation reste dans un cadre de recherches ou d'activités académiques. Il a été développé par l'équipe CISMef, CHU-Hôpitaux de Rouen, France. Très complet, il permet l'accès à plus de 65 terminologies et ontologies (T/O) concernant le domaine de la Santé. Ces dernières sont généralement spécifiques à un domaine précis de la Santé (maladies, anatomie, etc.).

### *Ontoserver*

Ontoserver [12] est un serveur multiterminologique clinique propriétaire basé sur la norme Fast Health Interoperability Resources (FHIR), développé par The Commonwealth Scientific and Industrial Research Organisation (CSIRO)<sup>8</sup>. Certaines de ses principales caractéristiques comprennent : la prise en charge immédiate des terminologies SNOMED CT, LOINC et Web Ontology Language (OWL) telles que l'ontologie du phénotype humain (HPO abréviation en anglais) ; un algorithme de recherche rapide basé sur un préfixe pour s'assurer que les utilisateurs peuvent facilement trouver du contenu et ne sont pas découragés d'entrer des données codées ; un mécanisme de syndication pour faciliter la mise à jour des terminologies ; et une mise en œuvre complète du langage ECL (Expression Constraint Language) de SNOMED CT, qui permet des analyses de données sophistiquées.

---

<sup>8</sup> <https://www.csiro.au/en/> (consulté le 21 janvier 2021)

### ***LexEVS (Lexical Grid Enterprise Vocabulary Services)***

LexEVS [33] est un serveur multiterminologique distribué sous une licence en code ouvert du EVS (Enterprise Vocabulary Services)<sup>9</sup> basé sur Java, développé par le National Cancer Institute (NCI). Il fournit un modèle terminologique commun et un accès ouvert à un large éventail de terminologies, d'ensembles de valeurs terminologiques et des mises en correspondance interterminologiques nécessaires à NCI et à ses partenaires.

Pour les utilisateurs souhaitant installer et exécuter une instance locale de LexEVS, le *package* du serveur LexEVS fournit un ensemble complet de logiciels et de services pour charger, publier et déployer du vocabulaire dans une variété d'environnements Web. L'API LexEVS sert les utilisateurs qui souhaitent un accès programmatique aux données mises à disposition par LexEVS.

### ***Snow Owl***

Snow Owl<sup>10</sup> est un serveur multiterminologique *en code ouvert* développé par B2i Healthcare [34]. Il propose une plateforme collaborative de création de terminologie et fournit l'API utilisée pour développer son navigateur (Snow Owl MQ)<sup>11</sup>. La plateforme de création gère des artefacts terminologiques développés par une équipe et pris en charge par des flux de travail d'entreprise pilotés par des systèmes de gestion des tâches externes tels que Bugzilla et JIRA. Grâce à sa conception modulaire, le serveur peut maintenir plusieurs terminologies où de nouvelles terminologies peuvent être connectées à la plateforme. Les fonctionnalités de Snow Owl sont exposées via une API REST. Les clients peuvent facilement accéder et interroger SNOMED CT, LOINC, ATC, ICD-10 et des dizaines d'autres terminologies via les API REST ou Java. La création distribuée collaborative est également prise en charge, y compris la création et la maintenance de

---

<sup>9</sup> <https://wiki.nci.nih.gov/display/lexevs/lexevs> (consulté le 04 février 2020)

<sup>10</sup> <http://b2i.sg/terminology-server/> (consulté le 10 décembre 2020)

<sup>11</sup> <https://mq.b2i.sg/snow-owl/#> (consulté le 30 novembre 2020)

systèmes de codification locaux, la mise en correspondance entre les terminologies, et la création de sous-ensembles de terminologie.

Snow Owl dispose d'une documentation Open API REST-API pour les terminologies prises en charge, notamment SNOMED CT et FHIR, etc. Snow Owl a une scène de développeur active, car B2i Healthcare publie simultanément des mises à jour sur serveur sur son référentiel git. Le logiciel public est sous licence Apache 2.0, mais la version exploitée en tant que service est propriétaire [1].

### ***BioPortal***

Développé par NCBO (National Center for Biomedical Ontology), BioPortal [35, 36] est un référentiel ouvert d'ontologies biomédicales qui permet d'accéder via des services web et des navigateurs à des ontologies développées en format Web Ontology Language (OWL), Resource Description Framework (RDF), Open Biological and Biomedical Ontology (OBO), Protégé et des terminologies. L'incorporation d'une variété de fonctionnalités web 2,0 permet au système de se comporter non seulement comme un référentiel d'ontologie complet, mais également comme une infrastructure générale pour prendre en charge l'accès communautaire, l'examen par les pairs, la mise en correspondance et l'annotation du contenu de l'ontologie.

BioPortal<sup>12</sup> fournit non seulement aux cliniciens et développeurs un guichet unique pour accéder par programmation aux ontologies biomédicales, mais intègre également les données de diverses ressources biomédicales. La technologie BioPortal est en code ouvert et est indépendante du domaine. Ainsi, d'autres communautés peuvent réutiliser le logiciel pour maintenir leurs propres référentiels d'ontologie [37].

---

<sup>12</sup> <https://bioportal.bioontology.org/> (consulté, le 15 mars 2020)

### **2.5.2 Présélection sur la base des besoins identifiés**

Aucun des cinq serveurs ne répond pleinement aux besoins retenus. La possibilité de déterminer les termes et les codes adéquats en fonction de la variété terminologique des sources lors de la rédaction du PAD n'est satisfaite par aucun SMT ainsi que la possibilité de traduire et gérer la traduction des termes et définitions et étiquettes, etc.

Or deux de ceux-ci n'étant pas en code ouvert, HeTOP et Ontoserver, il ne se sera pas possible de les modifier pour les rendre conformes à nos besoins. En conséquence, ils sont retirés de la liste. Les serveurs retenus pour l'évaluation se limitent donc à LexEVS, Snow Owl et BioPortal.

# Chapitre 3

## Méthodes d'évaluation

Ce chapitre traduit les besoins définis précédemment en exigences et pose les critères retenus aux fins de l'étude comparative des SMT considérés. La méthode résultante permettra de comparer les SMT en six étapes qui exploreront (1) l'utilisabilité (2) le modèle (3) les fonctionnalités des API et des IPM (4) la mise en correspondance sémantique et structurelle (5) la sécurité (6) les autres critères. Ces étapes ont été divisées en sous-étapes afin de faciliter l'examen conjoint de fonctionnalités apparentées.

### 3.1 Évaluation de l'utilisabilité

La norme ISO9241-11:2018 (fr) [38] définit l'utilisabilité comme le « degré selon lequel un système, un produit ou un service peut être utilisé, par des utilisateurs spécifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié ». Dans cette étape, nous reprenons une partie de l'étude réalisée par Wassing [1], mais en prenant en compte d'autres aspects avec des serveurs différents.

Pour les besoins du GRIIS, l'utilisabilité des SMT est évaluée selon la documentation, le support, les dépendances externes et la facilité d'utilisation. Pour qu'un SMT franchisse cette étape, il faut d'abord pouvoir le déployer complètement, puis il faut que l'interface de son API soit exposée et puisse être interrogée.

#### 3.1.1 Processus d'évaluation

Pour évaluer l'utilisabilité des serveurs retenus, nous avons utilisé deux euristiques d'utilisabilité proposées par Nielsen [39].

La documentation examine les fonctionnalités, le déploiement et l'utilisation, tout cela constitue des informations cruciales pour la maintenance et la configuration des serveurs. Les fonctionnalités décrivent les API disponibles et les terminologies prises en charge, le déploiement décrit comment mettre en place un serveur en état de fonctionnement et l'utilisation décrit comment utiliser les API, charger les fichiers de version des terminologies.

Pour évaluer la facilité de déploiement, nous avons défini un critère portant sur les dépendances externes. En effet, la nécessité d'installer des logiciels supplémentaires pour que le SMT fonctionne est considérée comme un facteur atténuant pour le déploiement. Un grand nombre de dépendances signifie généralement beaucoup plus de temps investi dans la lecture de documents, la mise en place d'environnements et la configuration de logiciels. Par conséquent, moins de dépendances signifient une meilleure convivialité. Nous avons considéré un serveur avec les moins de dépendances comme étant le meilleur.

Nous avons considéré un grand nombre des dépendances si l'installation exige plus des trois dépendances.

Afin de passer à l'étape suivante, un serveur doit être déployé. Un serveur est considéré comme déployé une fois qu'il peut être interrogé via l'API REST dans un navigateur et renvoyé des données des terminologies.

Nous avons transposé chaque euristique en termes de besoin et choisi un sous-ensemble d'objets relativement auxquels des critères sont définis pour l'évaluation dans le Tableau 2 ci-après.

**Tableau 2 — Critères d'utilisabilité**

<b>Besoin</b>	<b>Objet</b>	<b>Critères évalués</b>
Aide et documentation	Documentation	c1 : Existence. c2 : Organisation. c3 : Exhaustivité.
Efficacité d'utilisation	Facilité de déploiement	c1 : Minimalité des dépendances externes c2 : Minimalité des paramètres de configuration

## 3.2 Évaluation du modèle

Dans cette étape, l'objectif est de faire sortir les modèles de bases données de chaque serveur afin de savoir s'il permet de répondre aux différentes requêtes à effectuer. Un modèle de données est défini comme une représentation schématique d'un ensemble de données terminologiques, de leurs relations et de leurs traductions linguistiques. Des modèles sont utilisés pour stocker des terminologies, ils peuvent être relationnels, hiérarchiques ou autres. Le modèle à évaluer doit être capable de fournir une base pour créer des API cohérentes et standardisées, pour accéder aux multiples terminologies prenant en charge les requêtes de recherche, la navigation dans la hiérarchie et un ensemble riche de fonctionnalités. Le principe d'un SMT est de prendre en charge plusieurs terminologies. Ceci est facilité par un modèle de terminologie commun qui définit comment les terminologies sont formatées et représentées par programme, et est censé être suffisamment flexible pour représenter avec précision une grande variété de terminologies développées dans différents langages et d'autres ressources de connaissances telles que les ontologies.

Le modèle doit fournir une représentation de base pour toutes les données gérées et récupérées via l'API, et peut se représenter les terminologies fournies dans de nombreux formats sources, y compris les OBO, OWL, XML, etc.

Une fois que des informations des terminologies peuvent être représentées dans un modèle standard, il devient possible de créer des terminologies pour stocker du contenu ainsi que de créer des API pour accéder et manipuler ce contenu.

Pour évaluer le(s) modèle(s) du serveur, nous allons comparer les composants essentiels de chaque modèle du serveur par rapport aux autres qui sont retenus en cherchant les ressemblances et les différences existant entre eux. L'architecture sur laquelle les modèles sont basés sera examinée pour connaître les différentes actions qui peuvent être effectuées avec les serveurs.

Les critères d'évaluation sont synthétisés au tableau suivant.

**Tableau 3 — Critères d'évaluation du modèle**

<b>Besoin</b>	<b>Objet</b>	<b>Critères évalués</b>
Représentation terminologique	Modèle terminologique de référence Modèle relationnel interne correspondant	c1 : Cohérence du modèle de référence. c2 : Exploitabilité pratique du modèle relationnel.
Interopérabilité terminologique	Modèle relationnel interne	c1 : Capacité de tenir compte différentes terminologies c2 : Capacité d'exprimer les principaux modèles terminologiques (plus particulièrement ceux utilisés dans le domaine de la santé) selon le modèle de référence.
Intégration du multilinguisme au modèle	Modèle terminologique de référence Modèle relationnel interne correspondant	c1 : Identification explicite de la langue des éléments textuels du modèle. c2 : Modélisation de la multiplicité linguistique.

### 3.3 Évaluation des fonctionnalités des API et des IPM

Une fois le serveur déployé et les modèles évalués vient l'étape de l'évaluation des API et des fonctionnalités des IPM ou, autrement dit, la capacité et la performance du SMT à accomplir les différentes fonctionnalités.



En ce qui concerne l'évaluation des API, il est à noter que chaque SMT peut avoir ses propres API, dans ce cas, l'objectif est d'expérimenter les API génériques pour explorer les terminologies proposées par chaque serveur, nous nous intéressons ici sur l'utilisabilité d'une API générique, c'est-à-dire utiliser une même API pour consulter les terminologies. Il est intéressant de démontrer ici la capacité du serveur de définir de nouvelles terminologies propres à des acteurs particuliers. Nous allons aussi examiner la structure des objets de retour de points de terminaison en analysant les API REST telles que décrites par les serveurs, en plus d'examiner à travers les API les différentes requêtes qui peuvent être effectuées et les résultats renvoyés pour chaque requête reçue, afin de s'assurer que différentes fonctionnalités sont offertes par le serveur.

Pour évaluer les fonctionnalités des IPM, nous avons défini des fonctionnalités représentant les besoins ainsi que les services dans la section 2.4. Ceux-ci vont être utilisés pour ensuite tester le serveur. Chaque serveur sera évalué sur sa prise en charge d'un ensemble de fonctionnalités qui permet de créer et de gérer du contenu terminologique, cela inclut la capacité d'importation et d'exportation des terminologies.

Cette étape est le cœur de notre étude, parce que c'est ici qu'on examine la capacité du serveur à interagir et échanger des données avec d'autres serveurs ou applications comme cela aurait lieu en utilisant PARS3.

### **3.3.1 Spécifications des exigences des API et des IPM**

Le SMT doit répondre à un ensemble des exigences fonctionnelles et non fonctionnelles.

#### ***Exigences fonctionnelles***

Les exigences qui en découlent des besoins recensés auparavant sont les suivantes :

**F1** : le serveur doit faciliter le développement et l'utilisation des terminologies en permettant la gestion d'une terminologie (construction d'une terminologie lors de mise en correspondance, modification et suppression).

**F2** : permettre la gestion des contenus d'une terminologie (création, modification et suppression des concepts).

**F3** : permettre la recherche et l'exploration des terminologies.

**F4** : permettre la gestion de mise en correspondance des terminologies (création, modification, suppression et consultation de mise en correspondance entre des terminologies).

**F5** : permettre la gestion de mise en correspondance des contenus des terminologies (création, modification, suppression et consultation des mises en correspondance entre les concepts des terminologies).

**F6** : permettre le recensement des catalogues des terminologies par source (création, modification, suppression, consultation des mises en correspondance entre une source, une ontologie et une terminologie).

**F7** : offrir un système d'authentification à travers un identifiant et un mot de passe pour identifier les utilisateurs qui souhaitent accéder aux terminologies.

**F8** : permettre le contrôle d'accès ou la gestion des autorisations pour vérifier les droits attribués aux utilisateurs et les tâches qu'ils peuvent exécuter, etc.

**F9** : permettre l'importation et l'exportation des terminologies et de leurs contenus.

**F10** : Être capable de montrer l'ensemble des codifications utilisées par chacune des sources de données et leurs définitions.

**F11** : une interface conviviale est nécessaire.

**F12** : les terminologies doivent être disponibles pour toute personne impliquée dans le système de PARS3.

### ***Exigences non fonctionnelles***

En plus des exigences fonctionnelles, nous avons défini d'autres exigences qui agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur d'où leurs importances. Pour cela, le SMT doit répondre aux exigences non fonctionnelles suivantes :

**NF1** : Fiabilité : le serveur doit fonctionner de façon cohérente sans erreurs.

**NF2** : Les erreurs : le serveur doit les signaler par des messages d'erreurs.

**NF3** : Efficacité : le serveur doit permettre l'accomplissement de la tâche avec le minimum de manipulations.

**NF4** : Sécurité : le serveur doit être sécurisé : authentification et contrôle d'accès.

**NF5** : Capacité évolutive : le serveur doit être adaptable à toutes les évolutions organisationnelles.

### **3.3.2 Synthèse des critères**

Les critères d'évaluation sont synthétisés au tableau suivant.

**Tableau 4 — Critères d'évaluation des API et des IPM**

<b>Besoin</b>	<b>Objet</b>	<b>Critères évalués</b>
Pouvoir gérer les terminologies : création, suppression et lecture des terminologies.	Terminologie	c1 : Création, c2 : Suppression, c3 : Lecture des terminologies
Pouvoir gérer les concepts : création, mise à jour, suppression et lecture des concepts (définitions, etc.)	Concept	c1 : Création, c2 : Suppression, c3 : Lecture des concepts
Pouvoir gérer mettre à jour des terminologies sur la base de sources externes	Terminologie Source externe	c1 : Mise à jour des terminologies
Pouvoir traduire et gérer la traduction des termes et définitions et étiquettes	Terminologie Concept	c1 : Traduire un terme c2 : Traduire une définition c3 : Traduire une étiquette c4 : Modifier une traduction
Rechercher au sein d'une terminologie	Terminologie Concept	c1 : Rechercher un terme par un code c2 : Rechercher textuellement des concepts
Obtenir une liste des concepts d'une terminologie	Terminologie Concept	c1 : Liste des concepts c2 : Consulter les détails
Afficher la structure hiérarchique d'une terminologie et pouvoir y naviguer	Terminologie	c1 : Afficher le détail d'une terminologie c2 : Naviguer
Pouvoir importer des terminologies dans différents formats standards OWL, XML, etc.	Terminologie	c1 : Importer une ressource
Pouvoir exporter des terminologies dans différents formats standards OWL, XML, etc.	Terminologie	c1 : Exporter une ressource

### 3.4 Évaluation de l'alignement et de la liaison

La liaison sémantique et structurelle est un alignement entre une terminologie, une ontologie et une source de données, c'est-à-dire, pouvoir lier les éléments d'une source avec sa terminologie correspondante ainsi que l'ontologie utilisée ou bien montrer qu'un

champ de l'Ontorel (schéma relationnel de l'ontologie de PARS3) pour une source donnée c'est la terminologie d'un tel est dans tel serveur.

Dans cette étape, nous allons évaluer les mécanismes d'alignement proposés par chaque serveur, c'est-à-dire la capacité des SMT à lier avec exactitude les concepts considérés comme identiques, mais provenant de différentes terminologies, ontologies et sources :

- ◇ **Ajout des alignements** : permettre de créer ou d'insérer des alignements entre des concepts d'une terminologie, une ontologie et une source.
- ◇ **Visualisation des alignements** : permettre d'afficher les alignements par une IPM ou une API.
- ◇ **Téléchargement des alignements** : permettre de transférer les alignements du serveur vers un autre ou vers une application pour une utilisation antérieure.

Les critères d'évaluation sont synthétisés au tableau suivant.

**Tableau 5 — Critères d'évaluation de l'alignement et de la liaison**

Besoin	Objet	Critères évalués
Pouvoir mettre en correspondance les concepts d'une terminologie (intraterminologique)	Liste des concepts d'une terminologie mis en correspondance	c1 : Ajouter des alignements c2 : Visualiser les alignements c3 : Modifier les alignements c4 : Supprimer des alignements c5 : Télécharger des alignements
Pouvoir mettre en correspondance les concepts des terminologies différentes (interterminologique)	Liste des concepts d'une terminologie mis en correspondance	c1 : Ajouter des alignements c2 : Visualiser les alignements c3 : Modifier les alignements c4 : Supprimer des alignements c5 : Télécharger des alignements
Pouvoir mettre en correspondance les sources, les ontologies et les terminologies (recensement des terminologies par source).	Liste des sources, des ontologies et des terminologies mises en correspondance	c1 : Ajouter des alignements c2 : Visualiser les alignements c3 : Modifier les alignements c4 : Supprimer des alignements c5 : Télécharger des alignements

### **3.5 Évaluation de la sécurité**

La cinquième étape évalue la sécurité, un aspect important puisque les SMT peuvent contenir des informations sensibles qu'il faut pouvoir protéger tout en garantissant l'intégrité des données et un certain niveau de disponibilité. Les fonctions de sécurité de chaque serveur seront ainsi évaluées sur leur capacité à :

- ◇ offrir un système d'authentification à travers un identifiant et un mot de passe pour identifier les utilisateurs qui souhaitent accéder aux terminologies ;
- ◇ contrôler l'accès ou gérer des autorisations pour vérifier les droits attribués aux utilisateurs et les tâches qu'ils peuvent exécuter.

Les critères d'évaluation sont synthétisés au tableau suivant.

**Tableau 6 — Critères d'évaluation de la sécurité**

<b>Besoin</b>	<b>Objet</b>	<b>Critères évalués</b>
Utiliser un système d'authentification	Authentification	c1 : Authentification : identifier les utilisateurs qui demandent l'accès c2 : Permettre l'utilisation d'un service d'authentification externe (OpenLDAP <sup>13</sup> , SAML <sup>14</sup> ou OAuth <sup>15</sup> )
Permettre le contrôle d'accès	Ressources Utilisateurs	c1 : Autorisation : pouvoir établir l'accès aux ressources et aux fonctionnalités par utilisateur c2 : Traçabilité : pouvoir tracer l'utilisation des ressources et des fonctionnalités par utilisateur

### 3.6 Évaluation des autres critères

Dans la dernière étape, nous évaluerons les critères correspondant à des besoins variés qui ne se classaient pas dans les thèmes explorés dans les cinq étapes précédentes. Il s'agit des éléments suivants :

- ◇ **Contenu** : désigne la partie du serveur contenant les terminologies échangées entre les utilisateurs, les différences notables sur certaines fonctionnalités du contenu pouvant montrer les forces et les faiblesses de chaque outil seront notées ici. Ce critère est composé des caractéristiques suivantes :
  - **Source de T/O** : définit l'origine des terminologies et ontologies (T/O), un serveur fait appel à différentes sources pour obtenir des T/O. Ces sources proviennent des responsables de T/O, de leurs mises à jour, des T/O ajoutées localement par les utilisateurs du serveur.

---

<sup>13</sup> Open-source LDAP (lightweight directory access protocol)

<sup>14</sup> Security Assertion Markup Language (SAML)

<sup>15</sup> Open Authorization (OAuth)

- **Formats pris en charge** : définit la structure de données termino-ontologiques contenues sur un serveur, établie des règles qui régissent le stockage, l’affichage et la manipulation de ces données.
- **Métadonnées des T/O** : c’est une caractéristique formelle normalisée et structurée, utilisée pour la description et le traitement des contenus des ressources termino-ontologiques.
- **Métriques** : indique le nombre d’éléments intégrés (T/O, concepts, synonymes, définitions, etc.).
- ◇ **Architecture et technologies utilisées** : cette caractéristique désigne la structure générale du serveur et l’ensemble des techniques utilisées pour mettre en place un serveur.
  - **Architecture** : cette caractéristique désigne la structure générale du serveur, l’organisation des différents éléments du serveur (systèmes d’exploitation, logiciels ou matériels, etc.) et des relations entre les éléments.
  - **Technologies utilisées** : l’ensemble des outils et des techniques utilisés pour réaliser un serveur.
  - **Bases de données** : est un « conteneur » stockant des données termino-ontologiques.
- ◇ **Licence** : définit les droits d’utilisation du serveur. La licence recherchée doit permettre minimalement la modification et la redistribution du code aux mêmes conditions.
- ◇ **Disponibilité du code source** : le serveur doit avoir un code source facilement accessible sans aucune restriction.
- ◇ **Extensibilité** : Le serveur doit avoir une architecture modulaire permettant d’intégrer de nouveaux modules et d’autres outils pour la faire évoluer en fonction des besoins demandés.
- ◇ **Outil de gestion** : Le serveur doit disposer d’un ensemble complet d’outils de programmation et gestion tels qu’une interface en ligne de commande et une interface web utilisateur. La combinaison de ces deux types d’interfaces permet aux



développeurs d'utiliser des scripts pour les interactions automatisées à travers la ligne de commande et de gérer facilement leurs applications via l'interface web tout en réduisant la complexité des tâches de développement.

Les critères d'évaluation sont synthétisés au tableau suivant.

**Tableau 7 — Autres critères d'évaluation**

<b>Besoin</b>	<b>Objet</b>	<b>Critères évalués</b>
Capacité de gérer et de stocker plusieurs terminologies	Contenu	c1 : Sources des terminologies c2 : Formats pris en charge c3 : Métadonnées des terminologies c4 : Métriques
Présentation de la structure du serveur	Architecture	c1 : Architecture du serveur c2 : Technologies utilisées c3 : Base de données du serveur c4 : Extensibilité
Capacité d'installer et d'utiliser le serveur	Licence Code source	c1 : Type de licence c2 : Disponibilité du code du serveur c3 : Possibilité d'installation locale c4 : Logiciels requis pour installation
Capacité d'offrir des outils	Outil de gestion	c1 : Interface de ligne de commande c2 : Interface graphique

# Chapitre 4

## Étude comparative des serveurs considérés

Ce chapitre présente les résultats de l'évaluation des trois SMT retenus selon les critères de comparaison définis au Chapitre 3.

Pour évaluer les critères et les fonctionnalités nous avons utilisé : « OUI » pour marquer la disponibilité ou la satisfaction de l'élément évalué, « NON » pour le cas contraire tandis que « S.O. » (sans objet), « N.E. » (non évaluable) marque que l'élément évalué est non applicable ou pas de réponse.

### 4.1 Évaluation de l'utilisabilité

Nous décrivons ici la qualité de documentation des serveurs ainsi que la facilité de déploiement.

#### 4.1.1 Documentation

##### *LexEVS*

La documentation de LexEVS se trouve sur le site de NCI<sup>16</sup> et sur le référentiel GitHub. La documentation est souvent peu précise et demande de multiples renvois. L'organisation de la documentation est bonne, mais est un peu dispersée, surtout quand il s'agit de l'utilisation des API, il faut cheminer de page en page pour réunir l'information requise. Toutes les fonctionnalités importantes de LexEVS se trouve dans la page de confluence spécifique pour les fonctionnalités<sup>17</sup>. Un guide de déploiement de LexEVS décrit les

---

<sup>16</sup> <https://wiki.nci.nih.gov/display/LexEVS/LexEVS> (consulté le 04 février 2020)

<sup>17</sup> <https://wiki.nci.nih.gov/display/LexEVS/LexEVS+6.5.4+Functionality+Overview> (consulté le 25 juin 2020)

configurations prises en charge et les instructions techniques. Les documents décrivant l'utilisation de LexEVS se trouvent principalement dans deux sections, une section guide d'administration qui décrit la configuration de l'environnement du point de vue d'une installation existante et une section guide du programmeur qui explique l'API LexEVS (services, utilitaires...) ainsi que de nombreuses API connexes. Dans l'ensemble, la documentation de LexEVS est complète, cependant, certains documents sont obsolètes et renvoient aux anciennes versions de LexEVS. Au moment de l'écriture de ces lignes, la dernière version de LexEVS était 6.5.4.

**Tableau 8 — Résumé de la documentation de LexEVS**

<b>Documentation</b>	Existence	Organisation	Exhaustivité
Fonctionnalités	OUI	OUI	OUI
Déploiement	OUI	OUI	OUI
Utilisation	OUI	NON	NON

### ***Snow Owl***

La documentation de Snow Owl se trouve sur plusieurs endroits. Tout d'abord sur la page d'accueil<sup>18</sup>, en suite sur la page de documentation [40] et le dépôt GitHub. La documentation de Snow Owl voit des améliorations et ne cesse de s'améliorer. Du fait que la documentation est un travail en cours pendant cette étude, il est difficile d'obtenir des informations nécessaires pour mettre en place rapidement Snow Owl. Les fonctionnalités de Snow Owl ont été bien décrites sur le dépôt GitHub. Cependant, des liens qui répertorient des informations obsolètes ou inexistantes dans le dépôt Git ont été l'obstacle majeur pour le déploiement ainsi pour tester correctement les API, malgré cela nous avons pu réaliser le déploiement de Snow Owl. L'utilisation de Snow Owl est documentée sur la page d'accueil et grâce à cela il était possible de comprendre les appels d'API.

---

<sup>18</sup> Snow owl Accueil, <https://docs.b2i.sg/snow-owl> (consulté le 01 décembre 2020)

**Tableau 9 — Résumé de la documentation de Snow Owl**

<b>Documentation</b>	Existence	Organisation	Exhaustivité
Fonctionnalités	OUI	OUI	NON
Déploiement	OUI	NON	NON
Utilisation	OUI	NON	NON

### ***BioPortal***

La documentation principale de BioPortal se trouve sur la page wiki<sup>19</sup> de NCBO [41]. L'organisation de la documentation est bonne, car elle est correctement cataloguée. Les fonctionnalités de BioPortal incluent la possibilité de parcourir, de rechercher et de visualiser les ontologies. Par ailleurs, BioPortal est beaucoup plus fréquemment documenté par des tiers que les deux autres serveurs.

La partie utilisation des appels d'API existe sous la forme de la documentation d'API REST et se trouve sur la page API Documentation<sup>20</sup>. Elle est très riche et plus complète que celles LexEVS et Snow Owl.

Un guide de déploiement d'une application virtualisée de BioPortal (OntoPortal) existe et décrit les configurations prises en charge et les instructions techniques.

**Tableau 10 — Résumé de la documentation de BioPortal**

<b>Documentation</b>	Existence	Organisation	Exhaustivité
Fonctionnalités	OUI	OUI	NON
Déploiement	OUI	OUI	NON
Utilisation	OUI	OUI	OUI

---

<sup>19</sup> NCBO Wiki, [https://www.bioontology.org/wiki/Main\\_Page](https://www.bioontology.org/wiki/Main_Page) (consulté le 19 juin 2021)

<sup>20</sup> BioPortal REST API, <http://data.bioontology.org/documentation> (consulté le 19 juin 2021)

## ***Synthèse des résultats d'évaluation de la documentation***

Une documentation riche est disponible pour les trois serveurs, cependant elle est plus organisée et exhaustive pour BioPortal et LexEVS.

### **4.1.2 Facilité de déploiement**

#### ***LexEVS***

Le déploiement de LexEVS était plutôt facile grâce à la documentation fournie et aux échanges par courriel avec la National Institutes of Health (NIH) de NCI. Pour une installation standard (c'est-à-dire lorsqu'installé à partir d'un exécutable fourni par NCI et déployé en tant que serveur local), il n'y a que quatre dépendances. Il s'agit du système de gestion de base de données MySQL 6.5, du connecteur MySQL5.1.6 (une version supérieure ou inférieure n'est pas acceptée), de java 8 (ou supérieur) et d'un conteneur Apache Tomcat 8.5. Bien que d'autres configurations soient possibles, elles n'ont pas été testées. Les utilisateurs souhaitant télécharger le code source de LexEVS seront soumis à d'autres dépendances telles que Eclipse 4.6.x, jdk 1.8, Apache Ant 1.8.x, Maven 3.x et Git 1.9.x).

La documentation de l'API REST de LexEVS se limite à la fonction de recherche GET, conséquemment seule cette fonction a été testée. LexEVS fournit d'autres API tels que LexBIG [42] et Common Terminology Services 2 (CTS2), mais l'API LexBIG n'a pas été testée, faute de temps.

**Tableau 11 — Déploiement pour LexEVS**

<b>Facilité de déploiement</b>	Minimalité des dépendances externes	Minimalité des paramètres de configuration
Installation standard	NON	NON
Installation à partir de sources	NON	NON

## ***Snow Owl***

Après avoir éprouvé des problèmes à déployer Snow Owl parce que la documentation de déploiement était insuffisante, nous avons eu recours aux foires aux questions de GitHub pour arriver à nos fins.

Dans l'ensemble, la documentation de Snow Owl couvre pour l'essentiel, mais néglige plusieurs détails incontournables au déploiement. Par ailleurs, plusieurs liens sont obsolètes ou renvoient à des pages inexistantes.

Snow Owl exige un certain nombre de conditions préalables, si la construction est pour le développement, il sollicite une éclipse avec un certain nombre de configurations et de plugiciels (plug-ins en anglais).

Snow Owl a des avantages en tant que SMT parce qu'il a peu de dépendances, on a compté quasiment deux au total. Il n'a besoin que de java11. L'autre besoin étant le chargement des fichiers de terminologies pour fournir des résultats lors de l'interrogation.

Il est possible d'obtenir de l'aide de développeurs lorsqu'un problème survient, ce qui a été le cas pour nous en consultant les réponses aux problèmes GitHub soumis à ce sujet et autres sujets<sup>21</sup>.

**Tableau 12 — Déploiement pour Snow Owl**

<b>Facilité de déploiement</b>	Minimalité des dépendances externes	Minimalité des paramètres de configuration
Installation standard	OUI	OUI
Installation à partir de sources	OUI	OUI

---

<sup>21</sup> Snow owl issues, <https://github.com/b2ihealthcare/snow-owl/issues> (consulté le 16 juin 2021)

## ***BioPortal***

BioPortal offre une copie nommée OntoPortal, qui est une application virtualisée (Appliance) de BioPortal et qui peut être installé localement avec une licence enregistrée.

La documentation de la version actuelle de OntoPortal (V 3.0.0) se concentre sur l'administration de l'application virtualisée. Compte tenu de cette documentation, le déploiement de OntoPortal était facile en suivant les étapes d'installation qui se varient en fonction de la plateforme utilisée. Après l'installation, il faut s'inscrire pour obtenir un certificat de licence qui est gratuit pour les étudiants ou pour une utilisation individuelle. De plus OntoPortal fonctionne dans des environnements de virtualisation tels que VMware et VirtualBox. Il peut également fonctionner avec KVM (Kernel-based Virtual Machine), mais cela n'a pas été testé.

Après l'installation, le serveur était pleinement opérationnel, car un portail pour gérer les terminologies et les ontologies était fourni ainsi tous les points de terminaison de l'API REST étaient accessibles pour les tests. Néanmoins, les ontologies sont stockées dans la base de données centrale de BioPortal.

BioPortal ne fournissant pas ses sources, une installation à partir des sources n'est pas possible.

**Tableau 13 — Déploiement pour BioPortal**

<b>Facilité de déploiement</b>	Minimalité des dépendances externes	Minimalité des paramètres de configuration
Installation standard	OUI	OUI
Installation à partir de sources	N. E.	N. E.

## ***Synthèse des résultats d'évaluation de déploiement***

Le déploiement complet de chaque serveur a été réalisé, néanmoins Snow Owl facilite plus le déploiement avec un minimum des dépendances et des paramètres de

configuration. BioPortal ne permet pas un déploiement à partir de sources. Les deux types de déploiement pour LexEVS nécessitent plusieurs dépendances.

## 4.2 Évaluation du modèle

Dans cette section, nous présentons les résultats d'évaluation du modèle de données des trois SMT étudiés.

### 4.2.1 Description du modèle de chaque serveur

#### *LexEVS*

LexGrid [43] est le modèle de données de LexEVS, et est en fait un ensemble de sous-modèles. En effet, il est composé de plus de 8 sous-schémas définissant chacun un domaine (CodingSchemes, Relations, Concepts, Value Set Definition, Pick List Definition, etc.). Si l'on veut appréhender LexGrid en entier, il faut gérer pas moins de 80 entités reliées entre elles. Ces entités permettent la gestion fine de certains éléments (versions, types de données, types de relations, etc.).

Ce modèle est multiterminologique et multilingue, car il permet de stocker plusieurs terminologies dans différentes langues. Il répond exactement aux besoins d'un SMT.

Pour charger une terminologie dans LexEVS, l'information requise est seulement URI (Uniform Resource Identifier) de la terminologie. Cependant, une terminologie peut être créée en format OWL, XML, OBO ou autres et importer dans LexEVS.

Voici quelques-uns des objets de niveau supérieur incorporés dans la définition du modèle de LexEVS :

- ◇ **CodingSchemes** — aussi appelé système de codification, chaque terminologie est modélisée comme un système de codification unique, qui pourrait être considéré



comme un conteneur pour les concepts, les relations, les propriétés et les métadonnées.

- ◇ **Entity** — chaque Entity peut être de type « Concept », « Instance », « Association », etc.
- ◇ **Concept** — sous-classe de l'Entity.
- ◇ **Association** — sous-classe de l'Entity et représente la relation entre les concepts.
- ◇ **Instance** — sous-classe de l'Entity.
- ◇ **Property** — association avec le CodingSchemes, Entity ou Association.
- ◇ **Value Set Definition** — aussi appelé ensemble de valeurs, spécifie un ensemble de codes tirés d'un ou plusieurs systèmes de codification, destinés à être utilisés dans un contexte particulier. Les jeux ou ensembles des valeurs peuvent être sélectionnés à partir de différents systèmes de codification.
- ◇ **Pick List Definition** — définit une liste ordonnée de codes d'entité et de désignations préférées tirées de Value Set Definition résolue.

### ***Snow Owl***

Snow Owl stocke ses terminologies dans une base de données centrale de B2i Healthcare. En conséquence, il ne documente pas son modèle de données. Il donne cependant un aperçu de haut niveau du modèle conceptuel qui comprend les sous-modèles suivants :

- ◇ **Terminology** (CodingSchemes) — la terminologie est la structure qui définit et encapsule l'ensemble de composants terminologiques (entités) d'un système de codification (par exemple, un ensemble de codes avec leur signification) ainsi que ses différentes versions. Une terminologie est identifiée par un nom unique et stockée dans un référentiel. Plusieurs systèmes de codification peuvent exister dans un même référentiel les uns à côté des autres tant que leur nom est unique.
- ◇ **Terminology Component** : Un composant terminologique est un élément de base dans un système de codification ayant une signification ou une utilisation clinique

réelle. Par exemple, dans SNOMED CT, le concept, la description, la relation et le membre de l'ensemble de référence sont des composants terminologiques.

- ◊ **Version** : Une version fait référence à un instantané important dans le temps, cohérente à travers de nombreux composants de terminologie, également appelée balise ou étiquette. Elle est souvent créée lorsque l'état de la terminologie est jugé prêt à être publié et distribué aux clients en aval ou pour un usage interne. Une version est identifiée par son ID de version (ou balise de version) dans un système de codification donné.

D'autres entités, voire des sous-modèles, sont également mentionnées : Repository, Revision et Branch.

### ***BioPortal***

BioPortal ne repose pas sur un méta-modèle, mais sur une collection d'ontologies stockées dans un entrepôt RDF. BioPortal utilise LexGrid [43] pour stocker les T/O en format OBO et Rich Release Format (RRF), et Protégé<sup>22</sup> [44] pour les autres formats. Du fait de leur politique qui consiste à n'intégrer que des T/O aux formats ontologiques, les imports sont largement facilités. Quelques ajustements peuvent cependant avoir lieu si certains éléments ne sont pas bien représentés. BioPortal ne gère pas le multilinguisme, mais propose des « vues » sur des versions des T/O dans d'autres langues. Il existe encore quelques anomalies à ce sujet, car le NCBO travaille actuellement sur une intégration d'autres langues que l'anglais. Comme Snow Owl, BioPortal n'a pas un schéma de base de données importable et toutes les données sont hébergées sur leur base de données centrale.

---

<sup>22</sup> <https://protege.stanford.edu> (consulté le 26 juin 2021)

## Synthèse des résultats d'évaluation

Tableau 14 — Synthèse d'évaluation du modèle

Critères	LexEVS	Snow Owl	BioPortal
Cohérence du modèle de référence	OUI	OUI	OUI
Exploitabilité pratique du modèle relationnel	OUI	NON	NON
Capacité de tenir compte différentes terminologies	OUI	OUI	OUI
Capacité d'exprimer les principaux modèles terminologiques selon le modèle de référence	OUI	OUI	OUI
Identification explicite de la langue des éléments textuels du modèle	OUI	NON	OUI
Modélisation de la multiplicité linguistique	OUI	NON	NON

### 4.3 Évaluation des fonctionnalités des API et des IPM

La comparaison de fonctionnalités des API et des IPM se retrouve ici pour les trois serveurs évalués. En effet selon les besoins définis au Chapitre 2, nous avons défini une liste des fonctionnalités à évaluer.

Dans cette section, nous allons comparer les fonctionnalités à travers les requêtes des API et IPM. Cela nous permet de choisir un SMT qui offre plus des fonctionnalités dont on a besoin pour minimiser le cout.

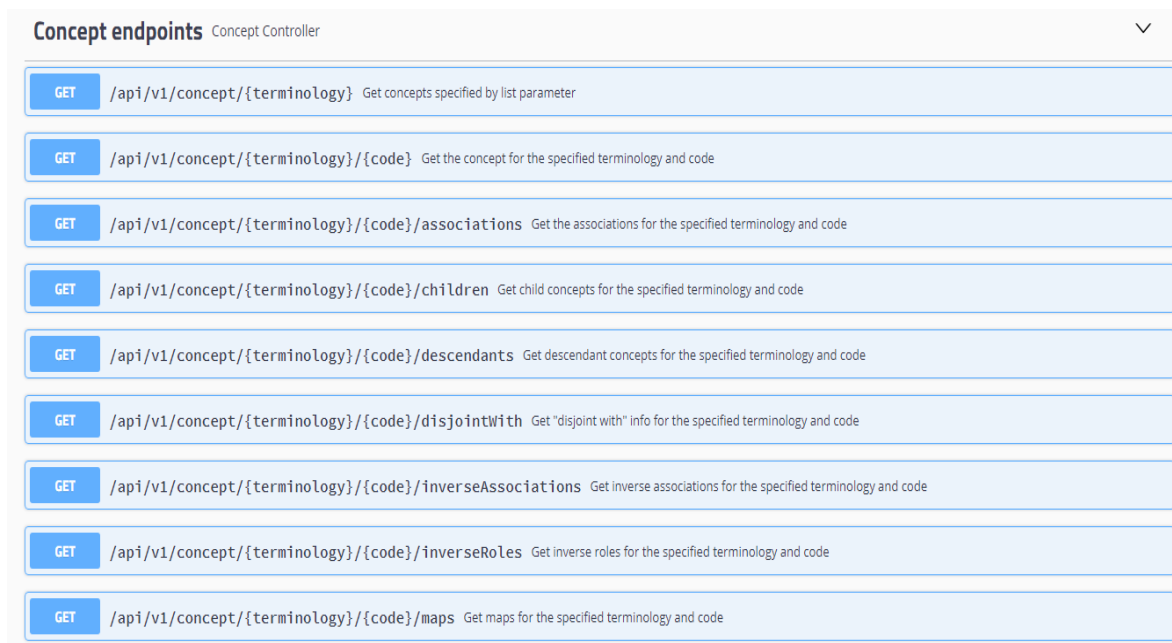
#### 4.3.1 Évaluation des API

En effet, nous avons résumé les résultats dans les récapitulatifs des tableaux dans les sections suivantes, ces résultats montrent la disponibilité ou non des fonctionnalités comparées à travers des réponses aux requêtes API ou IPM, cependant, nous n'avons pas montré les résultats à travers des captures d'écran des interfaces pour les IPM et des codes Json et XML retournés par les API pour ne pas rendre le document trop volumineux.

## LexEVS

LexEVS fournit plusieurs API, cependant nous n'en avons évalué que deux à savoir CTS2 et REST API. La documentation des API de LexEVS se limite seulement sur la recherche GET, donc nous avons testé ces API en nous basant sur la documentation fournie.

L'API CTS2 de LexEVS fournit un accès programmatique qui permet à l'implémentation des fonctionnalités dont les besoins les nécessitent. Il est possible de créer des applications, des outils personnalisés qui peuvent effectuer des appels contre l'API LexEVS CTS2. Mais les problèmes rencontrés lors de l'interrogation de cet API étaient que beaucoup de liens sont obsolètes. La Figure 6 et la Figure 7 illustrent les API REST et CTS2 de LexEVS.



Concept endpoints		Concept Controller	▼
GET	/api/v1/concept/{terminology}	Get concepts specified by list parameter	
GET	/api/v1/concept/{terminology}/{code}	Get the concept for the specified terminology and code	
GET	/api/v1/concept/{terminology}/{code}/associations	Get the associations for the specified terminology and code	
GET	/api/v1/concept/{terminology}/{code}/children	Get child concepts for the specified terminology and code	
GET	/api/v1/concept/{terminology}/{code}/descendants	Get descendant concepts for the specified terminology and code	
GET	/api/v1/concept/{terminology}/{code}/disjointWith	Get "disjoint with" info for the specified terminology and code	
GET	/api/v1/concept/{terminology}/{code}/inverseAssociations	Get inverse associations for the specified terminology and code	
GET	/api/v1/concept/{terminology}/{code}/inverseRoles	Get inverse roles for the specified terminology and code	
GET	/api/v1/concept/{terminology}/{code}/maps	Get maps for the specified terminology and code	

**Figure 6 — Une partie de l'API REST de LexEVS.**

Cette partie spécifique met en évidence certaines opérations impliquant les concepts, par exemple : rechercher des concepts de terminologies via cette API, à condition qu'un code et une terminologie soient donnés. La création et la mise à jour des concepts ne sont pas disponibles pour cette API

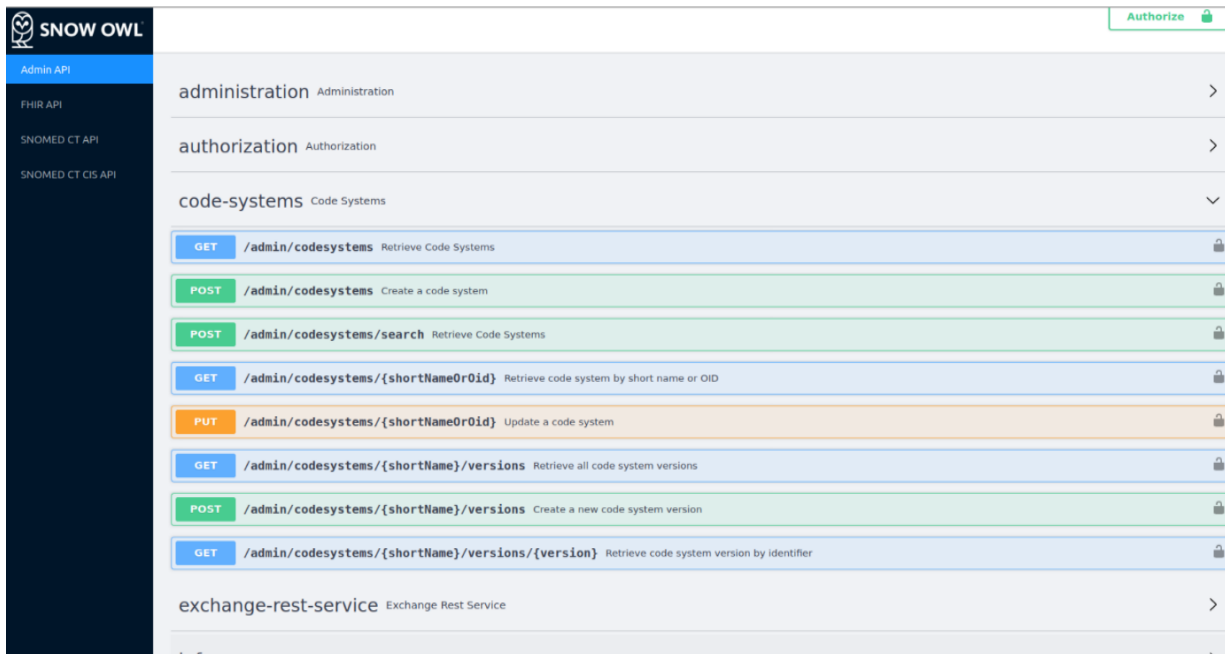
Service Metadata		<a href="#">List Methods</a>	<a href="#">Expand Methods</a>
GET	Get CTS2 Service Metadata	/service	
Entities		<a href="#">List Methods</a>	<a href="#">Expand Methods</a>
GET	Search Entities	/entities	
HEAD	Count Entities	/entities	
GET	Read an Entity by URI	/entitybyuri	
GET	Read an Entity by ID	/entity/{entityid}	
Code System Versions		<a href="#">List Methods</a>	<a href="#">Expand Methods</a>
GET	Search Code System Versions	/codesystemversions	
GET	Read a Code System Version by Version ID	/codesystem/{codesystem}/version/{version}	
GET	Search Entities of a Code System Version	/codesystem/{codesystem}/version/{version}/entities	
GET	Read an Entity of a Code System Version by ID	/codesystem/{codesystem}/version/{version}/entity/{entityid}	

**Figure 7 — Une partie de l'API CTS2 de LexEVS.**

Cette partie spécifique met en évidence certaines opérations impliquant les entités, les systèmes de codification, etc. par exemple : rechercher des systèmes de codification via cette API, à condition qu'un identifiant ou une version soit donné.

### ***Snow Owl***

Nous avons évalué les quatre API fournies par Snow Owl pour interroger le contenu des terminologies, ils s'agissent d'API FHIR, API Snomed, API Admin et API CIS, cependant et contrairement à LexEVS, nous avons évalué ces API à travers les requêtes GET, PUT et POST et les résultats correspondaient à ce que nous attendions. La Figure 8 montre l'un des API de Snow Owl, Admin API.



**Figure 8 — Une partie de l'API Admin de Snow Owl.**

Cette partie spécifique met en évidence certaines opérations impliquant les systèmes de codification, etc. par exemple : rechercher, créer et mettre à jour des systèmes de codification via cette API, à condition qu'un identifiant ou une version soit donné.

La documentation des API de Snow Owl est gérée par un navigateur, il subit des pertes de performances après une utilisation prolongée, il faut réactualiser la page, en revanche la conception d'API REST est centrée sur moins de points de terminaison, mais le serveur renvoie beaucoup d'informations par point de terminaison.

### ***BioPortal***

BioPortal fournit une API REST qui est composée d'un ensemble de ressources (ontologies, classes, etc.). L'utilisation des appels d'API existe sous la forme de la documentation d'API REST, elle est très riche et plus complète par rapport aux documentations d'API de LexEVS et Snow Owl. L'évaluation de cet API a été réalisée à travers les requêtes GET, PUT et POST et les résultats correspondaient à ce que nous attendions.

Resource collection: [/ontologies](#)

#### HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies
GET	/ontologies/:acronym
GET	/ontologies/:acronym/latest_submission
GET	/ontologies/:acronym/download
GET	/ontologies_full
PUT	/ontologies/:acronym
POST	/ontologies
PATCH	/ontologies/:acronym
DELETE	/ontologies/:acronym

**Figure 9 — Une partie de l'API REST de BioPortal**

Cette partie spécifique met en évidence certaines opérations impliquant les ontologies. Par exemple : rechercher, créer, supprimer et mettre à jour des ontologies via cet API.

## Synthèse des résultats d'évaluation

Tableau 15 — Synthèse d'évaluation des API

Critères	LexEVS	Snow Owl	BioPortal
Création, suppression des terminologies	NON	OUI	OUI
Lecture des terminologies	OUI	OUI	OUI
Création, mise à jour et suppression des concepts (définitions, etc.)	NON	OUI	OUI
Lecture des concepts (définitions, etc.)	OUI	OUI	OUI
Mise à jour des terminologies sur la base de sources externes	NON	NON	NON
Traduire un terme	NON	NON	NON
Traduire une définition	NON	NON	NON
Traduire une étiquette	NON	NON	NON
Modifier une traduction	NON	NON	NON
Rechercher un terme par un code	OUI	OUI	OUI
Rechercher textuellement des concepts	OUI	OUI	OUI
Liste des concepts	OUI	OUI	OUI
Consulter les détails	OUI	OUI	OUI
Afficher le détail d'une terminologie	OUI	OUI	OUI
Naviguer dans une terminologie	OUI	OUI	OUI
Importer une ressource	NON	OUI	OUI
Exporter une ressource	NON	OUI	OUI

### 4.3.2 Évaluation des IPM

Dans un premier temps, une remarque a été faite que les trois SMT possèdent plusieurs fonctionnalités communes, mais chacun d'entre eux a ses propres spécificités.

#### *Service de stockage*

Dans ce paragraphe, nous allons évaluer la capacité du SMT à stocker et administrer les terminologies.



**Tableau 16 — Évaluation des fonctionnalités d'importation (chargement)**

<b>Fonctionnalités</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Importer un système de codification	OUI	OUI	OUI
Importer un jeu de valeurs	OUI	OUI	OUI
Importer un alignement	OUI	OUI	OUI

Le Tableau 16 permet de conclure que l'importation est similaire pour les trois SMT, car ils offrent les mêmes fonctionnalités. Cependant, pour charger les systèmes de codification, des jeux de valeurs ou d'alignements, ceux-ci doivent être créés dans un éditeur spécifique et respecter certains formats tels que OWL, OBO, XML, etc. avant d'être chargés dans le SMT.

Le chargement peut être effectué via une IPM ou ligne de commande pour LexEVS, en revanche pour Snow Owl et BioPortal nous n'avons pas pu tester le chargement par ligne de commande parce que le service n'était pas offert par les deux.

**Tableau 17 — Évaluation des fonctionnalités d'exportation**

<b>Fonctionnalités</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Exporter un système de codification	OUI	OUI	OUI
Exporter un jeu de valeurs	OUI	OUI	OUI
Exporter un alignement	OUI	OUI	OUI

Au niveau de l'exportation, nous pouvons conclure que les trois SMT proposent quasiment les mêmes caractéristiques. LexEVS a obtenu un score plus élevé, car, les IPM, et les requêtes en ligne de commande ont donné le même résultat, tandis que BioPortal et Snow Owl ne fournissent pas le service de requêtes en ligne de commande, mais leurs IPM fournissent le même résultat.

### ***Service de consultation***

Ce paragraphe présente les fonctionnalités de recherche qui sont évaluées par rapport aux résultats des IPM. Les résultats sont récapitulés aux Tableau 18 et Tableau 19.

**Tableau 18 — Évaluation des fonctionnalités de consultation d’alignement**

<b>Fonctionnalités</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Consulter l’alignement entre des systèmes de codification	OUI	OUI	OUI
Consulter l’alignement entre les concepts des systèmes de codification	OUI	OUI	OUI

Selon les résultats du tableau, les trois SMT satisfont les deux fonctionnalités mentionnées. Nous allons expliquer dans la section service de gestion comment les alignements sont créés par les trois SMT.

**Tableau 19 — Évaluation des fonctionnalités de consultation de SC**

<b>Fonctionnalités</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Consulter les systèmes de codification	OUI	OUI	OUI
Obtenir les versions disponibles d’un système de codification	OUI	OUI	OUI
Consulter les concepts d’un système de codification	OUI	OUI	OUI
Obtenir la liste des concepts	OUI	OUI	OUI
Obtenir les détails d’un concept	OUI	OUI	OUI
Consulter un concept en différentes langues	OUI	NON	NON

La consultation du système de codification (de son contenu et de ses versions) est offerte par les trois SMT, néanmoins la consultation d’un concept dans différentes langues faisant défaut pour BioPortal et Snow Owl.

**Tableau 20 — Évaluation des fonctionnalités de consultation de CU**

Fonctionnalités	LexEVS	Snow Owl	BioPortal
Obtenir le système de codification du contexte d'utilisation	OUI	S.O.	S.O.
Obtenir le concept du contexte d'utilisation	OUI	S.O.	S.O.
Obtenir le concept du contexte d'utilisation correspondant à un nom	OUI	S.O.	S.O.
Lister tous les concepts de contexte d'utilisation	OUI	S.O.	S.O.
Lister tous les identifiants de contexte d'utilisation	OUI	S.O.	S.O.

BioPortal et Snow Owl ne fournissant pas la possibilité de consulter de contexte d'utilisation, LexEVS est le seul SMT parmi les trois à offrir cette possibilité.

**Tableau 21 — Évaluation des fonctionnalités de consultation de JV**

Fonctionnalités	LexEVS	Snow Owl	BioPortal
Lister un jeu de valeurs d'un système	OUI	OUI	OUI
Obtenir les détails d'un jeu de valeurs	OUI	OUI	OUI
Lister le contenu d'un jeu de valeurs	OUI	OUI	OUI

Les fonctionnalités pour consulter ou lister les jeux de valeurs sont satisfaites par les trois SMT.

### ***Service de gestion***

Cette section présente les résultats pour les fonctionnalités de gestion, cela implique la gestion des systèmes de codification, d'alignement et jeux de valeurs, etc.

**Tableau 22 — Évaluation des fonctionnalités de gestion d’alignement**

Fonctionnalités	LexEVS	Snow Owl	BioPortal
Créer un nouvel alignement entre les concepts	OUI	OUI	NON
Créer un nouvel alignement entre des systèmes de codification	OUI	OUI	NON
Ajouter des qualificatifs à l’alignement	OUI	S.O.	NON
Déterminer des systèmes de codification source et cible pour l’alignement	OUI	OUI	NON
Mettre à jour un alignement	OUI	OUI	NON
Supprimer un alignement	OUI	OUI	NON

Les IPM de LexEVS et Snow Owl offrent la possibilité de gérer les alignements (créer, modifier ou supprimer l’alignement). Cependant, la gestion d’alignement se fait par un éditeur externe ensuite charger le schéma dans LexEVS, quant à Snow Owl cela est seulement possible à travers une interface fournie à cet effet.

**Tableau 23 — Évaluation des fonctionnalités de gestion de SC**

Fonctionnalités	LexEVS	Snow Owl	BioPortal
Créer un nouveau système de codification	OUI	OUI	OUI
Ajouter une nouvelle propriété au système de codification	OUI	NON	NON
Créer un nouveau concept dans un système de codification	OUI	OUI	NON
Ajouter une nouvelle propriété au concept	OUI	OUI	NON
Modifier des métadonnées d’un système de codification	NON	NON	OUI
Modifier un concept	OUI	OUI	NON
Supprimer un système de codification complet	OUI	OUI	OUI
Supprimer un concept	OUI	OUI	NON
Mettre à jour d’un système de codification	OUI	OUI	OUI

Nous avons analysé dans cette partie la gestion des systèmes de codification et ses contenus. LexEVS obtient un score élevé grâce à sa satisfaction de ce critère alors que BioPortal n’offre que peu de possibilités à travers son IPM.

**Tableau 24 — Évaluation des fonctionnalités de gestion de CU**

<b>Fonctionnalités</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Créer un nouveau système de codification pour contenir le contexte d'utilisation	OUI	S.O.	S.O.
Modifier un contexte d'utilisation	OUI	S.O.	S.O.
Supprimer un contexte d'utilisation	OUI	S.O.	S.O.
Modifier un contexte d'utilisation	OUI	S.O.	S.O.
Activer un contexte d'utilisation	OUI	S.O.	S.O.
Désactiver un contexte d'utilisation	OUI	S.O.	S.O.

LexEVS offre la possibilité d'ajouter, de modifier et de supprimer manuellement par les utilisateurs via une interface utilisateur les contextes d'utilisation. Cependant, BioPortal et Snow Owl ne le permettent pas.

**Tableau 25 — Évaluation des fonctionnalités de gestion de JV**

<b>Fonctionnalités</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Créer un nouveau jeu de valeurs	OUI	OUI	OUI
Ajouter une nouvelle propriété au jeu de valeurs	OUI	NON	OUI
Modifier les métadonnées d'un jeu de valeurs	OUI	OUI	OUI
Modifier une propriété d'un jeu de valeurs	OUI	NON	OUI
Supprimer un jeu de valeurs	OUI	NON	NON
Supprimer une propriété d'un jeu de valeurs	OUI	NON	NON
Modifier un jeu de valeurs	OUI	NON	NON

LexEVS et BioPortal offrent la possibilité de créer et modifier les jeux des valeurs à travers les API, il est aussi possible à travers leurs IPM de les ajouter manuellement.

En termes de suppression, seul LexEVS donne la possibilité de supprimer pleinement des jeux de valeurs à travers son IPM. Quant à Snow Owl, la mise en œuvre est incomplète en plus de présenter plusieurs défauts.

## Synthèse des résultats d'évaluation

Tableau 26 — Synthèse d'évaluation des IPM

Critères	LexEVS	Snow Owl	BioPortal
Création, suppression et lecture des terminologies	OUI	OUI	OUI
Création, mise à jour, suppression et lecture des concepts (définitions, etc.)	OUI	OUI	NON
Mise à jour des terminologies sur la base de sources externes	NON	NON	NON
Traduire un terme	NON	NON	NON
Traduire une définition	NON	NON	NON
Traduire une étiquette	NON	NON	NON
Modifier une traduction	NON	NON	NON
Rechercher un terme par un code	OUI	OUI	OUI
Rechercher textuellement des concepts	OUI	OUI	OUI
Liste des concepts	OUI	OUI	OUI
Afficher le détail d'une terminologie	OUI	OUI	OUI
Naviguer dans une terminologie	OUI	OUI	OUI
Importer une ressource	OUI	OUI	OUI
Exporter une ressource	OUI	OUI	OUI

### 4.3.3 Synthèse d'évaluation des API et IPM

Les API et les IPM de trois SMT ne sont pas complètement adéquates aux besoins de GRIIS :

- ◇ au niveau des API, l'API REST de BioPortal vient en première position en satisfaisant plus des critères, suivi des API de Snow Owl, quant à LexEVS ses API se limitent à la recherche ;
- ◇ concernant l'IPM, LexEVS et Snow Owl offrent de meilleurs résultats, l'IPM de BioPortal est moins complète ;

- ◊ quant à l'équivalence entre API et IPM, aucun SMT ne satisfait à ce critère, car les résultats obtenus pour leurs API ne sont pas semblables à ceux des IPM.

## 4.4 Évaluation de l'alignement et de la liaison

### 4.4.1 Liaison sémantique et structurelle

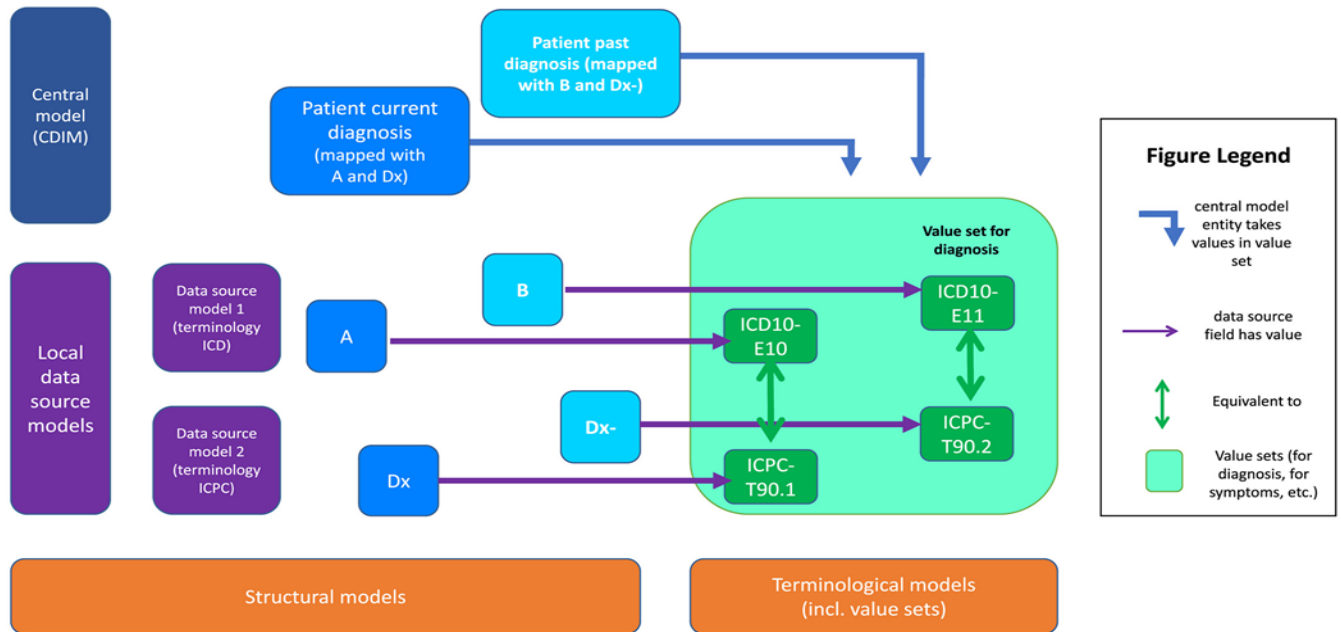
#### *LexEVS*

Des membres du GRIIS avaient déjà publié des résultats à ce sujet dans une étude où LexEVS a fait l'objet d'une double utilisation [17]. Premièrement, l'utilisation consistait à lier les codes entre différentes terminologies, avec des relations de synonymie ou de quasi-synonymie. Par exemple, deux terminologies ICPC-T90.2 et ICD10-E11 ont été mises en correspondance par une relation de synonymie, car elles font référence toutes deux au diabète sucré de type 2. Les codes ont été également regroupés dans des « ensembles de valeurs », qui sont des groupes de codes terminologiques, tels que la valeur définie pour le diagnostic. La Figure 10 illustre le résultat de cette étude.

Deuxièmement, l'utilisation consistait à mettre en correspondance un modèle d'ontologie avec chaque modèle de source de données, en mettant en relation les entités du modèle d'ontologie avec la combinaison de champs dans la source de données.

Enfin, LexEVS représentait la liaison sémantique/structurelle en liant les entités du modèle d'ontologie à des ensembles de valeurs. Par exemple dans la Figure 10, les entités « Patient current diagnosis » ou « Patient past diagnosis » ont été reliées à l'ensemble de valeurs pour le diagnostic. De plus, un champ de la base de données du modèle de la source de données 1, nommé Dx, peut contenir la valeur ICPC-T90. Étant donné que Dx représente un « Patient current diagnosis » et que le terme T-90 désigne un diabète non insulino-dépendant dans la Classification internationale des soins primaires 2 (ICPC-2 abréviation en anglais), il a été affirmé qu'il s'agit d'un diagnostic actuel de diabète non insulino-dépendant.

De manière générale, cette connaissance a été exprimée de la façon suivante : Dx est mis en correspondance à l'entité « Patient current diagnosis » de l'ontologie CDIM et ICPC-T90 est mis en correspondance à des entités synonymes dans d'autres terminologies telles que ICD10\_E10.



**Figure 10 — Utilisation de LexEVS pour une liaison structurelle/sémantique [17]**

En conclusion, LexEVS est le seul des SMT évalués qui permet de faire la liaison sémantique/structurelle selon les résultats de notre étude.

### ***Snow Owl et BioPortal***

Les SMT Snow Owl et BioPortal ne permettent pas d'établir de liaisons sémantiques/structurelles.

### **4.4.2 Alignements**

Les trois serveurs offrent des mécanismes d'alignements entre les terminologies et entre les concepts des terminologies selon les résultats suivants :



### ***LexEVS***

LexEVS offre la possibilité d'importer automatiquement les alignements, il est aussi possible à travers son interface utilisateur de les ajouter manuellement.

En termes de visualisation, LexEVS donne la possibilité de visualiser des alignements pour une terminologie ou un concept et les détails sur les alignements. Il est possible avec LexEVS d'exporter les alignements.

### ***Snow Owl***

Snow Owl permet une importation automatique d'alignement, et offre aussi la possibilité d'ajout manuel par les utilisateurs via une interface utilisateur. En plus, Snow Owl offre la capacité de visualiser des alignements pour les terminologies et les détails sur ces alignements. Il donne aussi l'occasion d'exporter ces alignements.

### ***BioPortal***

BioPortal permet une importation automatique (algorithmes et méthodes d'alignement), et offre aussi la possibilité d'ajout manuel par les utilisateurs via l'interface utilisateur. En termes de visualisation, BioPortal donne la possibilité de visualiser les alignements :

- ◇ **Par T/O** : un exemple pour les ontologies alignées avec SNOMED-CT est disponible à la page Mapping de SNOMED CT<sup>23</sup>. On peut y constater toutes les ontologies alignées avec SNOMED CT.
- ◇ **Par couple de T/O** : sur la même page, en sélectionnant deux T/O dans la liste des alignements, on peut consulter les alignements entre les deux. Finalement, en cliquant sur « **International Classification of Functioning, Disability and Health** », on peut visualiser en détail les 97 025 alignements entre les deux T/O.

---

<sup>23</sup> <https://bioportal.bioontology.org/ontologies/SNOMEDCT?p=mappings> (consulté le 21 juin 2021)

- ◊ **Par concept** : un exemple pour le concept « AIDS » de SNOMED-CT est disponible à la page Mapping<sup>24</sup>.

Enfin, les utilisateurs de BioPortal peuvent télécharger les alignements (sans pour autant télécharger des T/O).

### *Synthèse des résultats d'évaluation*

LexEVS permet une liaison sémantique et structurelle alors que Snow Owl et BioPortal ne la permettent pas, coté alignement les trois SMT offrent des résultats équivalents.

**Tableau 27 — Synthèse de l'alignement et de la liaison**

Critères	LexEVS	Snow Owl	BioPortal
Alignement intraterminologique	OUI	OUI	OUI
Alignement interterminologique	OUI	OUI	OUI
Alignement terminologie et source (Liaison sémantique et structurelle)	OUI	S.O.	S.O.

## 4.5 Évaluation de la sécurité

Dans cette partie, nous évaluons les fonctionnalités de chaque serveur pour savoir si le SMT est correctement sécurisé selon les exigences : authentification et contrôle d'accès.

### 4.5.1 Comparaison des SMT en fonction de la sécurité

#### *Snow Owl*

Snow Owl fournit des fonctionnalités qui permettent de sécuriser le serveur terminologique. Il offre la possibilité de protéger les données par un mot de passe et de mettre en œuvre des mesures de sécurité plus avancées telles que le contrôle d'accès. Par

---

<sup>24</sup> <https://bioportal.bioontology.org/ontologies/MEDDRA?p=mappings> (consulté le 21 juin 2021)

défaut, il est livré sans aucune fonctionnalité de sécurité activée. Cependant, il faudra configurer un domaine de sécurité.

Snow Owl fournit deux domaines qui sont : la configuration d'un domaine des fichiers et la configuration d'un domaine LDAP (Lightweight Directory Access Protocol).

Nous avons configuré le domaine LDAP parce qu'il offre plus de fonctionnalités, notamment l'autorisation des utilisateurs que le domaine des fichiers ne prend pas en charge.

Pour intégrer LDAP, nous avons configuré un domaine dans le fichier de configuration de Snow Owl.

- ◇ **Authentification** : après avoir configuré le domaine LDAP, Snow Owl a authentifié les demandes entrantes pour s'assurer que l'expéditeur de la demande est autorisé à accéder au serveur terminologique et à son contenu. Cependant, l'authentification se fait par un envoi du client d'un entête HTTP avec les valeurs utilisateur/mot de passe et Snow Owl envoyait une réponse HTTP 401 si une demande doit être authentifiée.
- ◇ **Contrôle d'accès ou autorisation** : nous avons configuré Snow Owl afin qu'il prenne en charge cette fonction ; ainsi, Snow Owl vérifiait si un utilisateur authentifié est autorisé à effectuer l'action demandée sur une ressource donnée. Dans Snow Owl, une autorisation est une valeur unique qui représente à la fois l'opération que l'utilisateur souhaite effectuer et la ressource à laquelle il accède. Le format est le suivant : « **opération** » : « **ressource** ». Les opérations prises en charge par Snow Owl :
  - **Browse**: lire le contenu d'une ressource.
  - **Edit** : écrire le contenu d'une ressource, supprimer une ressource.
  - **Import** : importer à partir de contenus et de formats externes.
  - **Export** : exporter vers du contenu et des formats externes.
  - **Version** : créer une version pour une terminologie.

Les ressources représentent le contenu auquel un client accède. Voici les formats de ressources dans Snow Owl :

- « **repositoryId** » : accéder à l'intégralité du contenu disponible dans un référentiel terminologique.
- « **repositoryId** » / « **branch** » : accéder au contenu disponible sur branche dans un référentiel terminologique.
- « **CodeSystemId** » : accéder à tout le contenu d'un système de codification.
- « **CodeSystemId** » / « **versionId** » : accéder à une version spécifique d'une terminologie.

Voici quelques exemples de l'autorisation :

- **Browse : snomedStore** — parcourir tous les systèmes de codification SNOMED CT et leur contenu.
- **Edit : SNOMEDCT** — modifier la terminologie SNOMED CT.
- **\* : SNOMEDCT** — permettre l'exécution de toutes les opérations sur la terminologie SNOMED CT.
- **Browse : \*** — autoriser les opérations de lecture sur toutes les ressources disponibles.
- **\* : \*** — autorisation administrateur, l'utilisateur peut faire n'importe quoi avec n'importe laquelle des ressources disponibles.

### ***BioPortal***

Un mécanisme de contrôle d'accès est mis en œuvre sur la base de la confidentialité des données et des privilèges d'accès des utilisateurs par BioPortal, ainsi pour empêcher les accès non autorisés, maintenir l'exactitude des données et assurer l'utilisation appropriée des informations, BioPortal met en place des procédures de gestion pour protéger les informations collectées en ligne.

L'authentification pour BioPortal se fait en fournissant un nom d'utilisateur et un mot de passe fournis lors de la création du compte sur le site de BioPortal. Ce compte permet à

son propriétaire de gérer ses propres terminologies, cependant des terminologies publiques sont disponibles à tout le monde même sans authentification.

De plus, il faut créer un compte sur BioPortal pour utiliser les services tels que les API, la soumission des nouvelles terminologies ou ontologies, etc. Néanmoins, BioPortal peut être utilisé sans un compte utilisateur, mais avec des services limités.

### ***LexEVS***

Les mécanismes d'authentification et de contrôle d'accès ne sont pas disponibles et n'ont pas pu être évalués pour LexEVS.

### ***Synthèse des résultats d'évaluation***

**Tableau 28 — Synthèse d'évaluation de la sécurité**

<b>Critères</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Authentification	S.O.	Mot de passe	Mot de passe
Service externe	S. O.	LDAP	NON
Autorisation	S.O.	OUI	OUI
Traçabilité	S.O.	OUI	OUI

## **4.6 Évaluation des autres critères**

Dans cette section, nous évaluons les autres critères qui ne sont couverts dans les parties précédentes, mais qui sont néanmoins pertinents dans le choix d'un serveur.

### **4.6.1 Contenu**

#### ***LexEVS***

LexEVS contient plus de 100 T/O axées essentiellement autour de ressources d'annotation (LOINC, CIM-9, Medical Dictionary for Regulatory Activities (MedDRA), SNOMED CT, Unified Medical Language System (UMLS), etc.) et pour la recherche

fondamentale en biologie moléculaire (Gene Ontology, ChEBI, etc.). La terminologie la plus mise en avant (sélection par défaut) est le NCI Thesaurus (NCIT) [25, 54] qui est le thésaurus développé par la communauté du Centre National du Cancer aux États-Unis. Il s'agit d'un référentiel en oncologie (et domaines transversaux) dont l'utilisation dépasse aujourd'hui les frontières américaines. Les différentes statistiques sur la volumétrie ne sont pas accessibles pour LexEVS, mais on peut estimer que le nombre de concepts terminologiques dépasse les 3 000 000.

### ***Snow Owl***

Snow Owl contient plus de 20 T/O qui sont définis dans les formats OWL, OWL2, Release Format 2 (RF2), Classification Markup Language (ClAML), etc. Les T/O sont importés de SNOMED CT, ATC, LOINC et IDC-10 qui sont largement utilisées à travers le monde pour différentes sortes d'applications. Les statistiques sur la volumétrie ne sont pas accessibles.

### ***BioPortal***

BioPortal contient plus de 930 T/O, ce qui correspond à plus 11 997 303 concepts et 48 ressources indexés. Le nombre des alignements exacts générés est 55 648 584 au moment d'écriture de ces lignes.

BioPortal permet d'importer des T/O dans différents formats tels que OWL, OBO, RDF ou bien Protégé Frame Language.

Le contenu du BioPortal est construit par ses utilisateurs. N'importe qui peut s'inscrire et soumettre sa propre T/O, ou apporter des commentaires ou des mises en correspondance pour des T/O déjà présentes. Alors que l'équipe de BioPortal effectue une curation légère des T/O, le projet a établi très peu de contraintes pour les soumissions des T/O. L'exigence minimale est que la T/O soit en quelque sorte pertinente pour le domaine de la biomédecine et qu'elle utilise l'un des formats pris en charge par BioPortal. Le domaine de l'informatique biomédicale est assez large et BioPortal contient des T/O qui

vont de l'anatomie, la description du phénotype, les conditions expérimentales, l'imagerie, la chimie à la santé [36].

### ***Synthèse des résultats d'évaluation***

Le tableau suivant synthétise les résultats d'évaluation du critère de contenu qui offre un avantage à LexEVS grâce à sa base de données qui peut être installée en local ce qui offre la possibilité de la modifier pour l'adapter aux besoins.

**Tableau 29 — Comparaison des contenus**

<b>Caractéristiques</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Propriétaire	NCI	B2i	NCBO
Source de terminologies	* UMLS * EVS  Autres sources ajoutées par les utilisateurs	* SNOMED CT * CIM10 * LOINC  Autres sources ajoutées par les utilisateurs	* UMLS * OBO Foundry  Autres sources ajoutées par les utilisateurs
Modèle de données	LexGrid	Méta-modèle multiterminologique	Entrepôt RDF, LexGrid, Protégé
Formats standards	OBO, OWL, OWL2, Text, UMLS, HL7 MIF, MedDRA, MetaData, NCI history, ICD9, ICD10, RRF, RIM database, Flat ...	OWL, OWL2, RF2, ClaML...	OWL, OBO, RDF, RRF, Protege Framework Language.
Métriques N (T/O) N (concepts) N (relations)	>100 > 3 000 000 ?	>20 ? ?	> 882 > 11 997 303 ?
Avantages	Base de données peut être installée en local		
Inconvénients	Schéma de données change après chaque ajout d'une terminologie	Base de données centralisée	Base de données centralisée

## **4.6.2 Architecture et technologies utilisées**

### ***LexEVS***

LexEVS est construit selon une architecture à N niveaux avec des interfaces client, des composants serveurs et des sources de données. Il est composé des couches suivantes :

- ◊ **La couche Application Service Layer** : cette couche accepte les demandes entrantes de toutes les interfaces publiques et les traduit, si nécessaire, en appels Java en termes de l'API LexEVS native,
- ◊ **La couche API de base** : sous-tend toutes les demandes d'API LexEVS,
- ◊ **La couche de source de données** : est responsable du stockage et de l'accès à toutes les données requises pour représenter les objets retournés via l'appel d'API.

LexEVS dispose d'un certain nombre de mécanismes d'API à utiliser avec diverses technologies, parmi elles : Java, JavaScript, Spring, Protege, LexGrid, etc. la majeure partie de l'infrastructure d'API LexEVS est écrite dans le langage de programmation Java.

MySQL 5.6 est la base de données prise en charge et préférée pour la version 6.5 de LexEVS.

MySQL 5.6 a été utilisé pendant le développement et dans les tests de LexEVS 6.5, et est également utilisé dans un environnement de production pour LexEVS 6.5. Bien que les autres bases de données puissent fonctionner pour LexEVS 6.5, elles n'ont pas été testées.

### ***BioPortal***

BioPortal adopte une approche d'architecture en couches, qui dissocie les modèles d'objets logiques et de domaine entre chaque couche :

- ◊ Une couche Présentation qui fournit l'interface utilisateur,
- ◊ Une couche Interface qui contient les Web services nécessaires pour les fonctionnalités de BioPortal,
- ◊ Une couche logique qui permet l'accès des API aux T/O, et aux Resources Index,
- ◊ Une couche Persistance pour les données.

Cette architecture tend à passer au modèle RDF triple store avec l'utilisation de 4store.



En termes de technologies, BioPortal utilise : Protege, LexGrid, Spring/JDBC, Hibernate, 4store.

### ***Snow Owl***

Snow Owl évolue d'un petit noyau intégré dans des produits mono-utilisateur à des clusters à N-tiers prenant en charge des centaines d'utilisateurs simultanés. Les clients peuvent facilement accéder et interroger les terminologies SNOMED CT, LOINC, ATC, ICD-10 et des dizaines des terminologies supplémentaires via les API REST ou Java.

**Tableau 30 — Comparaison de l'architecture et des technologies utilisées**

<b>Caractéristiques</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Architecture	Multicouches	Mono-Utilisateur à N-tiers	Multicouches
Technologies utilisées	Java, JavaScript, Spring, Protege, LexGrid, etc.	Java	Protege, LexGrid, Spring/JDBC, Hibernate, 4 store.
API REST	Oui, incomplète	Oui, incomplète	OUI
Portail disponible	OUI	OUI	OUI
Bases de données	MySQL, et autres	S.O.	S.O.

### **4.6.3 Autres caractéristiques**

Le tableau suivant résume les résultats des autres caractéristiques par rapport aux trois SMT étudiés.

**Tableau 31 — Comparaison des caractéristiques**

<b>Caractéristiques</b>	<b>LexEVS</b>	<b>Snow Owl</b>	<b>BioPortal</b>
Terminologies multilingues	Oui	Oui	Non
Navigation multilingue	Non (en)	Non (en)	Non (en)
License	Libre	Libre ou payante	Libre
Code	Ouvert	Ouvert	Fermé
Plateformes	CentOS 7.0, Fedora Core 22, Windows 10, macOS 10.x	Windows, Red Hat, Debian, Ubuntu, Conteneur Docker	Windows, MacOS, Linux
Logiciels requis pour installation	Java Runtime, MySQL 5.6, Apache Tomcat 8.5	Java 11 ou plus récent	VMware ou Virtual Box
Service centralisé	Oui	Oui	Oui
Installation locale possible	Oui	Oui	Oui

## 4.7 Analyse comparative

Cette recherche visait à évaluer les serveurs qui pourront interagir avec PARS3. Les critères d'évaluation proposés ont été testés avec un échantillon représentatif d'outils existants afin d'identifier les besoins actuels qui ne sont pas entièrement pris en charge pour définir, valider et gérer les terminologies utilisées dans le domaine de la santé.

### 4.7.1 Satisfaction des critères

#### *Utilisabilité*

Ce critère est complètement satisfait par les trois serveurs, mais avec un degré d'assistance différent. BioPortal possède la documentation la plus exhaustive, la plus accessible et son équipe de support est très réactive. La documentation de trois serveurs n'est disponible qu'en anglais. Relativement au déploiement, Snow Owl possède peu de dépendances (seul java 11 est requis) et requiert le chargement d'une seule

terminologie ; LexEVS facilite le déploiement avec un guide d'installation précis, mais demande l'installation de quelques composants logiciels au préalable. Le déploiement complet de chaque serveur a été réalisé et les tests des API et fonctionnalités ont pu être effectués. BioPortal, en raison de son approche par service uniquement, ne nécessite pas de déploiement.

### ***Modèle de données***

L'objectif de l'évaluation de ce critère était de savoir s'il est bien possible de changer la structure du schéma de la base de données du serveur pour pouvoir l'adapter aux besoins. Snow Owl dispose d'un modèle multiterminologique, mais qui n'est pas accessible à l'utilisateur, car il est basé sur un serveur central. LexEVS se positionne comme un standard terminologique via un modèle formel semi-complet par rapport aux besoins de GRIIS. Les deux modèles sont multiterminologiques, mais supportent des implémentations très différentes et donc, des investissements en temps différents. Du côté de BioPortal, le modèle n'est pas défini formellement ; il s'agit d'un entrepôt RDF et, comme toutes les ontologies, terminologies ou vocabulaires intégrés dans BioPortal possèdent des formats ontologiques standards, cela ne pose aucun problème. Le modèle est donc « caché » dans le code des fonctions dont certaines sont générales, mais d'autres spécifiques aux terminologies supportées. Cette approche ne se prête donc pas à la prise en compte de multiples terminologies de structures différentes.

L'approche LexEVS nécessite trop de temps pour appréhender le système et des fonctionnalités beaucoup trop contraignantes. Néanmoins, notre choix s'est porté sur le modèle LexEVS, car c'était le seul serveur à fournir son schéma de base de données, alors que les deux autres stockent les données dans un serveur central. LexEVS a obtenu le meilleur score grâce à sa satisfaction partielle de ce critère, mais la génération des tables périodiquement après chaque ajout d'une nouvelle terminologie nécessite d'avoir recours à la métaprogrammation qui, en plus d'être complexe, présente des risques

d'injections et offre des performances réduites. Par conséquent, aucun serveur ne satisfait pleinement ce critère.

### ***Fonctionnalités des API et des IPM***

Ce critère n'est pas satisfait. Si les SMT proposent chacun plusieurs des fonctionnalités requises, aucune ne permet de traiter le multilinguisme. De plus, les API de LexEVS ne permettent pas de créer, de supprimer ou de mettre à jour des ressources ; ils se limitent seulement à la lecture contrairement à son IPM qui offre plus des possibilités. Par ailleurs, les API et IPM de Snow Owl ne permettent pas de mettre à jour les terminologies sur la base des sources externes ainsi que la traduction des termes et des étiquettes, etc. Quant à BioPortal son IPM ne permet pas la création, la mise à jour, la suppression et la lecture des concepts (définitions, etc.). On remarque également que la couverture de l'API et celle de l'IPM sont différentes, et ce, pour les trois SMT. Toutefois, l'API REST de BioPortal pourrait être une source d'inspiration pour concevoir une nouvelle API plus complète.

### ***Liaison sémantique/structurelle***

Ce critère est satisfait seulement par LexEVS, car BioPortal et Snow Owl ne permettent pas de faire une liaison sémantique et structurelle.

### ***Sécurité***

Au niveau de l'authentification, LexEVS ne le permet pas alors que BioPortal et Snow Owl proposent des mécanismes d'authentification, toutefois seul Snow Owl permet d'utiliser un annuaire externe (LDAP). Au niveau de contrôle d'accès, Snow Owl est plus complet et plus flexible. Il donne une plus grande portée à la définition de droits d'accès. Il permet de définir les opérations que l'utilisateur souhaite effectuer et les ressources auxquelles il accède tandis que LexEVS et BioPortal ne les permettent pas. De manière globale, Snow Owl apporte quelques fonctionnalités indispensables à certaines utilisations, dont l'utilisation depuis l'écosystème PARS3.

### ***Autres critères***

Les critères que nous avons pu identifier et évaluer lors de cette étape partagent le plus souvent une évaluation semblable pour les trois serveurs avec un léger avantage pour LexEVS.

#### **4.7.2 Candidat le plus prometteur**

Après l'étude comparative des différents SMT, il en ressort que le candidat le plus avantageux parmi les SMT considérés c'est-à-dire celui qui correspond le mieux aux besoins des composantes de l'écosystème PARS3 et de ses utilisateurs est LexEVS. La suite du présent mémoire étudiera par conséquent comment intégrer les fonctionnalités manquantes : soit par la modification de LexEVS ou soit par l'écriture d'un SMT propre.

# Chapitre 5

## Estimation du projet de développement d'un nouveau SMT ou modification d'un SMT existant

Ce chapitre a pour objectif d'estimer l'envergure de l'effort requis pour mettre à disposition les fonctionnalités manquantes à un SMT apte à être utilisé dans un contexte de SSA et, plus particulièrement, lorsque couplé à PARS3. Sur la base du précédent chapitre, deux approches sont considérées : (1) modifier LexEVS (2) développer un SMT propre. Pour ce faire, deux techniques d'estimation seront utilisées : les points de fonctions (FP) et Constructive COst Model I (Cocomo-I).

### 5.1 Présentation des méthodes d'estimation utilisées

Il existe plusieurs modèles d'estimation d'efforts applicables aux projets de développement logiciel. Dans le contexte du présent mémoire, nous avons retenu deux méthodes éprouvées, complémentaires : Cocomo-I et FP.

#### 5.1.1 Cocomo

Cocomo est un acronyme pour Constructive COst Model [46]. C'est une méthode pour estimer le cout d'un projet logiciel dans le but d'anticiper l'envergure budget nécessaire et estimer les principaux jalons menant à la livraison.

Le premier modèle Cocomo a été développé en 1981 par Barry Boehm pour estimer le cout (en fonction d'un taux horaire déterminé), l'effort (en mois-personne), l'effectif optimal (en nombre de personnes) et la durée de développement (en mois) d'un sous-système logiciel dont la quantité de travail a été évaluée par ailleurs.

À partir de 2000, Boehm a développé Cocomo-II en adjoignant à Cocomo I une base multiparamétrique comprenant, notamment, le type de système, le degré de maturité de l'organisation, le taux de réutilisation de code, l'expérience antérieure de l'effectif, etc. Bien que plus précise, cette technique n'a pas été retenue ici, car il ne nous aurait pas été possible d'estimer la plupart de la centaine de paramètres requis faute de contexte précis de développement et de données historiques.

### **5.1.2 Points de Fonction (FP)**

Le point de fonction est une « mesure » exprimant l'envergure d'une fonctionnalité d'un sous-système exprimée en termes du domaine d'application et utilisable en pratique. À un point de fonction, est associée à une quantité de travail ; cette quantité de travail est ensuite convertie en effort. Plus simplement, les points de fonction sont utilisés pour calculer une mesure de taille fonctionnelle du logiciel et en dériver un effort utilisable dans le cadre d'une estimation grâce à une technique multiparamétrique, tels que Cocomo-I ou Cocomo-II.

Plusieurs techniques FP ont été développées au cours des ans par différentes organisations et divers groupes de chercheurs [47, 48]. L'Annexe A présente les détails de FP de la technique utilisée dans le présent mémoire.

## **5.2 Estimation des projets**

Ce processus correspond à l'application des modèles Cocomo-I et FP au projet du développement d'un nouveau SMT.

### **5.2.1 Présentation de l'outil d'estimation**

Pour rendre l'application des modèles plus conviviale et plus précise, nous avons créé un chiffrier automatisé. Ce chiffrier, présenté à la Figure 11, permet, d'une part des estimations de l'effort (PM, en mois-personnes), du travail en milliers de lignes de code « standardisées » (KSLOC), du cout (COST, en dollars), de la durée (TDEV, en mois) ainsi

le nombre optimal de personnes (P) et, d'autre part, des comparaisons entre les estimations de modèles utilisés.

<b>Présentation</b>				
Ébauche d'estimation de développement d'un nouveau serveur multi-terminologique				
<b>Paramètres d'organisation</b>				
SLOC/UFP	55,00	Pour la méthode FP-Cococo; 55 est une moyenne typique pour Java		
PM/UFP	0,20	Pour la méthode FP directe; 0,20 est un moyenne typique pour une organisation CMMI level = 2		
CAD/PM	7 000,00	Taux mensuel moyen tous postes confondus (incluant les couts d'exploitation)		
H/PM	141,00	Nombre moyen d'heures productives par mois et par personne		
<b>Paramètres de projet</b>				
(structure et complexité du cas mixte)				
Exigences	simple	moyen	complexe	
Intrant	13	4	6	
Extrant	4	0	0	
Requête	3	2	1	
S. interne	12	0	0	
S. externe	0	0	4	
Type de sy:	Couplé			
A	3,00			
E	1,12			
C	2,50			
F	0,35			
<b>Estimation</b>				
UFP	FP-simple	FP-moyen	FP-complexe	FP-mixte
Intrant	69	92	138	91
Extrant	16	20	28	16
Requête	18	24	36	23
S. interne	84	120	180	84
S. externe	20	28	40	40
Total (UFP)	207	284	422	254
FP directe				
PM	41,40	56,80	84,40	50,80
TDEV	9,20	10,28	11,81	9,89
P	4,50	5,53	7,15	5,14
COST	289 800,00	397 600,00	590 800,00	355 600,00
FP et Cococo-I				
KSLOC	11,39	15,62	23,21	13,97
PM	45,73	65,17	101,55	57,51
TDEV	9,53	10,79	12,60	10,32
P	4,80	6,04	8,06	5,57
COST	320 120,00	456 186,00	710 846,00	402 568,00
<b>Indicateurs</b>				
FP directe				
COST/M	31 492,58	38 679,82	50 035,77	35 972,40
FP et Cococo-I				
COST/M	33 596,79	42 294,76	56 428,49	38 993,28
KSLOC/Pl	0,25	0,24	0,23	0,24
<b>Paramètres Cococo-I</b>				
Type-sys	A	E	C	F
Autonomie	2,40	1,05	2,50	0,38
Couplé	3,00	1,12	2,50	0,35
Embarqué	3,60	1,20	2,50	0,32
<b>Paramètres FP</b>				
Catégorie	simple	moyen	complexe	
Intrant	3	4	6	
Extrant	4	5	7	
Requête	3	4	6	
S. interne	7	10	15	
S. externe	5	7	10	

Figure 11— Aperçu des résultats du chiffrier



Nous allons détailler ces résultats dans les sections suivantes.

### 5.2.2 Estimation du développement d'un nouveau SMT

Dans cette partie, nous allons, définir un nouveau modèle conceptuel sur lequel le nouveau SMT va se baser, identifier les exigences du nouveau système à développer et appliquer les deux modèles FP et Cocomo-I au projet de développement d'un nouveau SMT. Ce travail peut être schématisé de la façon suivante :

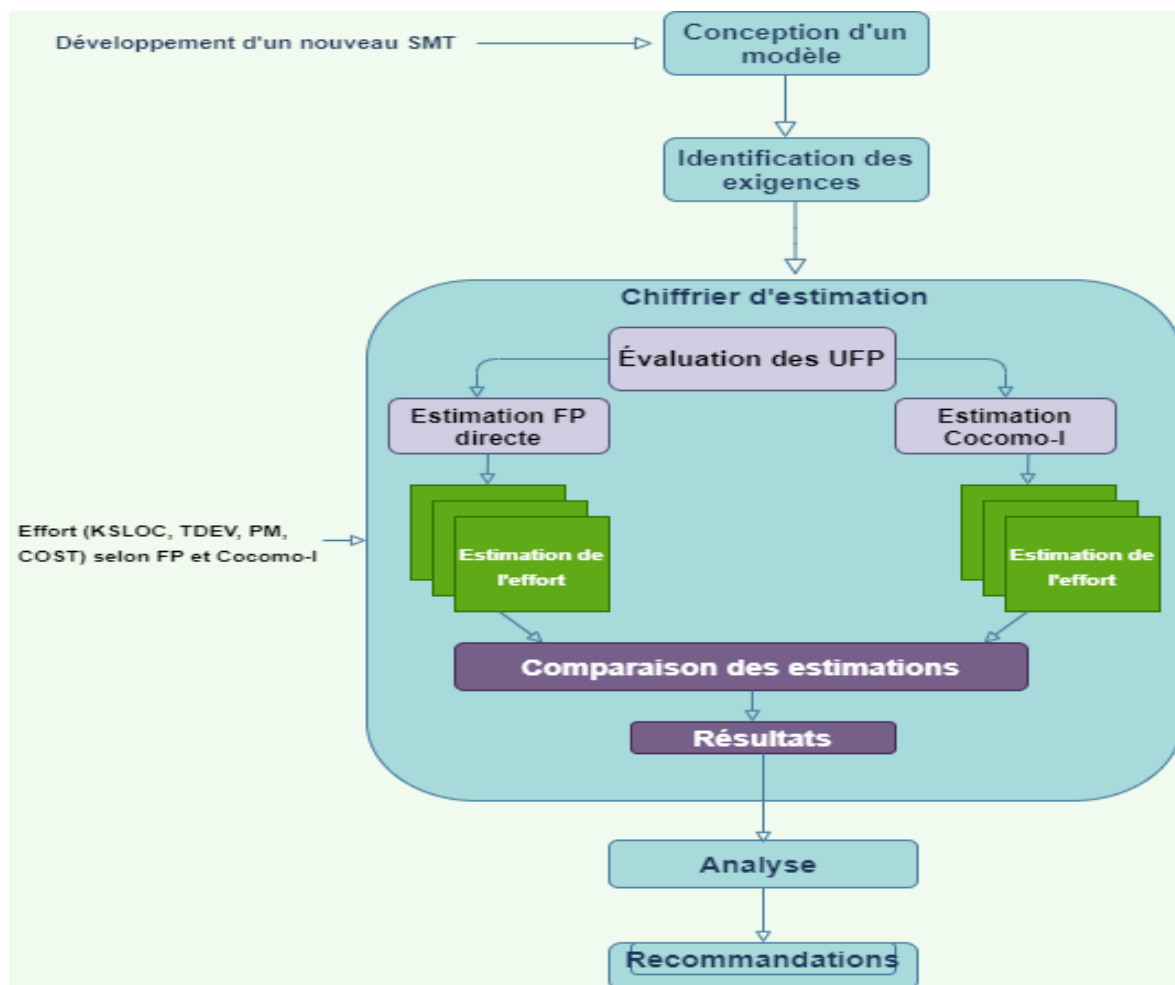


Figure 12 — Procédure d'estimation

### ***Nouveau modèle conceptuel de gestion des terminologies***

Nous avons défini un nouveau modèle conceptuel pour le développement d'un nouveau SMT. Notre approche pour modéliser un SMT repose sur la création d'un métamodèle terminologique, capable d'intégrer n'importe quel type de données terminologiques.

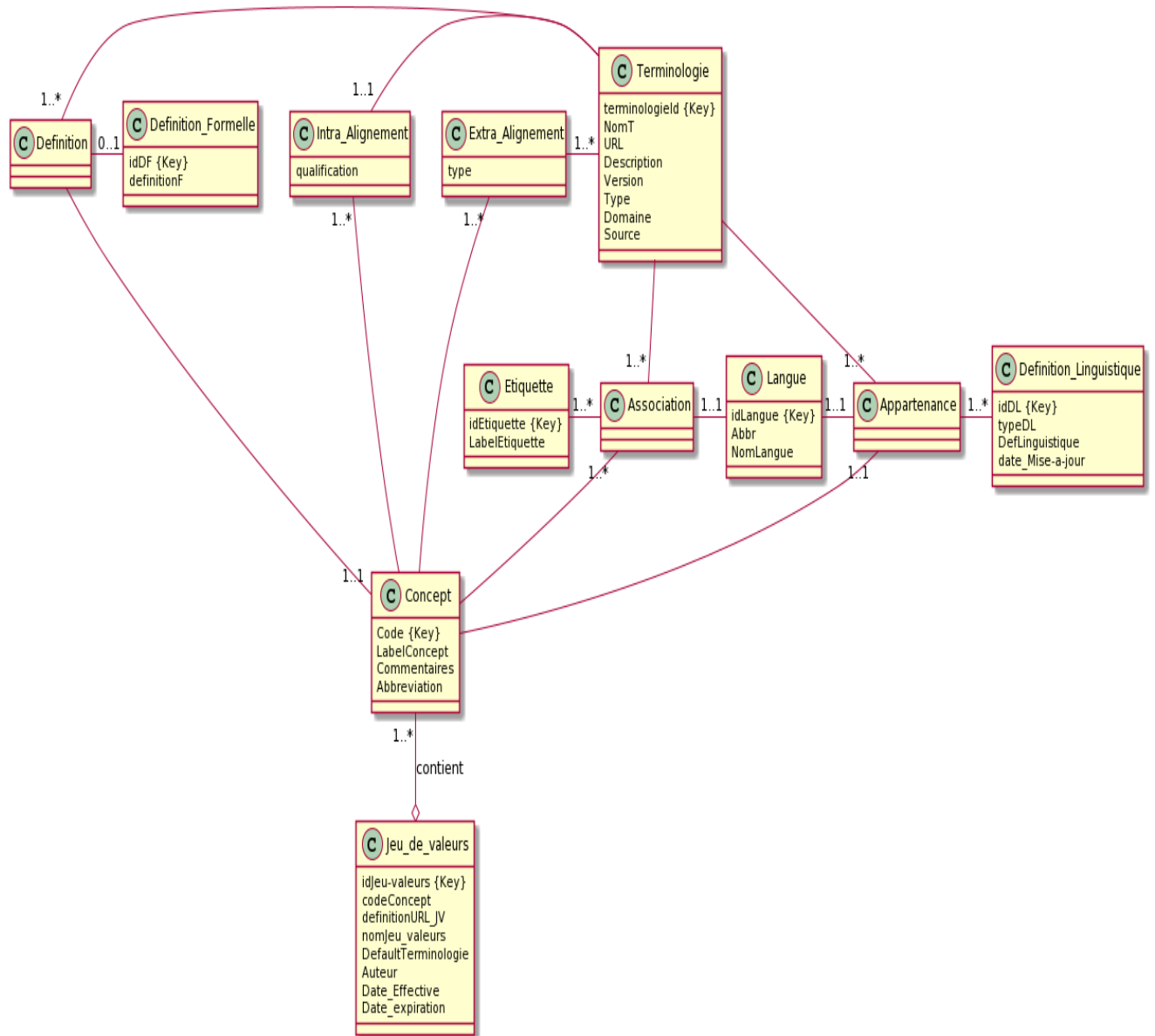
Afin de réunir dans une même structure, des terminologies, des thésaurus, des nomenclatures et des classifications de natures différentes ayant chacun des spécificités particulières, nous avons créé un méta-modèle qui tient compte de toutes ces particularités, et qui est assez générique pour avoir la possibilité d'intégrer d'autres terminologies. Pour se faire, le modèle Unified Modeling Language (UML) obtenu et décrit ci-dessous, voir la Figure 13.

Notre modèle définit au moins :

- ◇ Une Terminologie qui est unique, et est identifiée par `terminologieId`. Elle possède les attributs (Name, description, version, type et domaine). Une terminologie peut contenir un ou plusieurs concepts. Elle est associée à une source identifiée par un `idSource` et possédant un auteur et un titre.
- ◇ Un concept doit être unique dans une terminologie et posséder obligatoirement un identifiant nommé `code`, il possède un certain nombre d'attributs (Label, commentaire et abréviation). Il appartient à une et une seule terminologie, possède une définition linguistique dans une langue. Les concepts appartenant aux terminologies sont associés à une ou plusieurs étiquettes dans une langue. Une définition formelle d'un concept dans une terminologie est facultative.

Les concepts peuvent être mis en correspondance entre eux par deux types de relations :

- ◇ Les relations intraterminologiques qui impliquent deux concepts issus de la même terminologie. Les relations hiérarchiques font partie de ce type de relation.
- ◇ Les relations interterminologiques qui impliquent deux concepts issus de terminologies différentes.



**Figure 13 — Diagramme de classes des terminologies**

### *Identification des exigences*

Dans cette partie, nous identifions les exigences/composantes du nouveau SMT à développer, ainsi nous évaluons la complexité de chaque exigence afin de nous permettre de calculer les points de fonctions non ajustés (UFP).

Les deux tableaux ci-après montrent les différents éléments d'identification des exigences fonctionnelles (EF). Les tables incluent les transactions (Intrant (EI), Extrant (EO) et Requêtes (EQ)) et les données ou entités (Structure interne (ILF) et Structure externe (EIF)).

**Tableau 32 — Comptage des UFP d'EF (Gestion des terminologies)**

Transactions (EI, EO, EQ)	TE	Attributs	Dépôts référencés	Comp.	Valeur
Création d'une terminologie	EI	TerminologieId, nomT, URL, Description, Version, Type, Domaine, Source, Format, Licence, Publication	Terminologie	S	3
Modification d'une terminologie	EI	TerminologieId+ Attributs à modifier	Terminologie	S	3
Suppression d'une terminologie	EI	TerminologieId	Terminologie	S	3
Liste/Catalogue des terminologies	EQ	<b>En entrée :</b> terminologieId <b>En sortie :</b> toutes les terminologies	Terminologie	M	4
Consultation d'une terminologie	EO	TerminologieId	Terminologie	S	4
Recherche d'une terminologie	EQ	<b>En entrée :</b> terminologieId <b>En sortie :</b> la terminologie	Terminologie	M	4

**Notation**

TE : Type d'exigences

Comp. : Complexité

S : Simple

M : Moyenne

C : Complexe

**Tableau 33 — Comptage des UFP d'EF (Import/export de données)**

Transactions (EI, EO, EQ)	TE	Attributs	Dépôts référencés	Comp.	Valeur
Import d'une terminologie	EI		Terminologie, Concept	C	6
Export d'une terminologie	EQ	<b>En entrée :</b> terminologieId <b>En sortie :</b> Terminologie copiée	Tous	C	6

**Tableau 34 — Comptage des UFP d'EF (Gestion des langues)**

Transactions (EI, EO, EQ)	TE	Attributs	Dépôts référencés	Comp.	Valeur
Ajout d'une langue	EI	IdLangue, Abbr, nomLangue	Langue	S	3
Association d'une étiquette dans une langue	EI	IdÉtiquette, LabelÉtiquette, code, terminologieId, idLangue	Concept, Terminologie, Langue, Association, Étiquette	C	6

**Tableau 35 — Comptage des UFP d'EF (Gestion des concepts)**

Transactions (EI, EO, EQ)	TE	Attributs	Dépôts référencés	Comp.	Valeu
Création d'un concept	EI	Code, LabelConcept, Commentaires, Abréviation, terminologieId	Concept, Terminologie	M	4
Mise à jour d'un concept	EI	Code + Attributs à modifier	Concept	S	3
Suppression d'un concept	EI	Code	Concept	S	3
Recherche d'un concept	EQ	<b>En entrée :</b> code <b>En sortie :</b> Concept et tous ses attributs	Concept	S	3
Définition du code du concept	EI	Code, terminologieId	Concept, Terminologie, Definition	M	4
Ajout d'une définition linguistique à un concept	EI	IdDL, DefLinguistique, idLangue, code, terminologieId	Concept, Terminologie, Langue, Appartenance, Definition_Linguistique	C	6
Ajout d'une définition formelle à un concept	EI	IdDF, DefinitionF, code, terminologieId	Concept, Terminologie, Definition, Definiton_Formelle	C	6

**Tableau 36 — Comptage des UFP d'EF (Gestion d'alignement intraterminologique)**

Transactions (EI, EO, EQ)	TE	Attributs	Dépôts référencés	Comp	Valeur
Ajout des alignements entre les concepts d'une terminologie	EI	Code1, code2, terminologieId, qualification	Intra_Alignement, Concept, Terminologie	M	4
Modification des alignements entre les concepts d'une terminologie	EI	Code1, code2, terminologieId, qualification	Intra_Alignement, Concept, Terminologie	M	4
Visualisation des alignements entre les concepts d'une terminologie	EO	Qualification, code1, code2, terminologieId	Intra_Alignement	S	4
Suppression des alignements entre les concepts d'une terminologie	EI	Code1, code2, terminologieId, qualification	Intra_Alignement	S	3
Import d'alignements entre les concepts d'une terminologie	EI	Code1, code2, terminologieId, qualification	Intra_Alignement	S	3
Export d'alignements entre les concepts d'une terminologie	EQ	<b>En entrée :</b> code1, code2, terminologieId, qualification <b>En sortie :</b> Intra_Alignement copié	Intra_Alignement	S	3

**Tableau 37 — Comptage des UFP d'EF (Gestion d'alignement interterminologique)**

Transactions (EI, EO, EQ)	TE	Attributs	Dépôts référencés	Comp	Valeur
Ajout des alignements entre les concepts des terminologies différentes	EI	Code1, code2, terminologyID1, terminologyID2, type	Extra_Alignement, Concept, Terminologie	C	6
Modification des alignements entre les concepts des terminologies différentes	EI	Code1, code2, terminologyID1, terminologyID2, type	Extra_Alignement, Concept, Terminologie	C	6
Visualisation des alignements entre les concepts des terminologies différentes	EO	Code1, code2, terminologyID1, terminologyID2, type	Extra_Alignement	S	4
Suppression des alignements entre les concepts des terminologies différentes	EI	Code1, code2, terminologyID1, terminologyID2, type	Extra_Alignement	S	3
Import d'alignement entre les concepts des terminologies différentes	EI	Code1, code2, terminologyID1, terminologyID2, type	Extra_Alignement	S	3
Export d'alignement entre les concepts des terminologies différentes	EQ	<b>En entrée :</b> code1, code2, terminologyID1, terminologyID2, type <b>En sortie :</b> Extra_Alignement copié	Extra_Alignement	S	3



**Tableau 38 — Comptage des UFP d'EF (Gestion des jeux de valeurs)**

Transactions (EI, EO, EQ)	TE	Attributs	Dépôts référéncés	Comp.	Valeur
Création d'un jeu_de_valeurs	EI	IdJeu-valeurs, codeConcept, definitionURL_JV, nomJeu_valeurs, DefaultTerminologie, Auteur, Date_Effective, Date_expiration	Jeu de valeurs	S	3
Modification d'un jeu_de_valeurs	EI	IdJeu-valeurs, codeConcept, definitionURL_JV, nomJeu_valeurs, DefaultTerminologie, Auteur, Date_Effective, Date_expiration	Jeu de valeurs	S	3
Affichage d'un jeu de valeurs	EO	IdJeu-valeurs	Jeu de valeurs	S	4
Suppression d'un jeu_de_valeurs	EI	IdJeu-valeurs	Jeu de valeurs	S	3

**Tableau 39 — Comptage des UFP de Données (ILF, EIF)**

Données (ILF, EIF)	Type	Attributs	Comp.	Valeur
Terminologie	ILF	TerminologieId, nomT, URL, Description, Version, Type, Domaine, Source, Format, Licence, Publication	S	6
Concept	ILF	Code, LabelConcept, Commentaires, Abréviation, terminologieId	S	6
Jeu de valeurs	ILF	IdJeu-valeurs, codeConcept, definitionURL_JV, nomJeu_valeurs, DefaultTerminologie, Auteur, Date_Effective, Date_expiration	S	6
Intra_Alignement	ILF	Code1, code2, terminologieId, qualification	S	6
Extra_Alignement	ILF	Code1, code2, terminologyID1, terminologyID2, type	S	6
Association	ILF	IdÉtiquette, code, terminologieId, idLangue	S	6
Définition	ILF	Code, terminologieId	S	6
Définition formelle	ILF	Code, terminologieId, IdDF, definitionF	S	6
Définition linguistique	ILF	Code, terminologieId, idDL, typeDL, DefLinguistique, date_Mise_à_jour	S	6
Langue	ILF	IdLangue, nomLangue, Abbr	S	6
Appartenance	ILF	IdDL, code, terminologieId, idLangue	S	6
Étiquette	ILF	IdÉtiquette, LabelÉtiquette	S	6
Source de données arrimées avec PARS3	EIF		C	15
OWL	EIF		C	15
OBO	EIF		C	15
XML	EIF		C	15

Pour le calcul des points de fonction, les structures externes existantes (EIF) sont les modèles des sources de données arrimées avec PARS3 et les formats d'échange utilisés par données de données terminologiques externes dont le SMT est susceptible de tirer

des mises à jour terminologiques. Fort heureusement, du côté de PARS3 ce modèle est unique. Du côté des bases de données terminologiques, le nombre pourrait être considérable — dans la présente étude nous nous limiterons à trois (OWL, OBO, XML). Donc quatre structures différentes en tout, mais toutes complexes.

### **Calcul du nombre brut de FP**

Le calcul a été réalisé en sommant pour chaque type d'exigence fonctionnelle, les exigences ayant même niveau de complexité. Chaque nombre est ensuite multiplié par le poids correspondant. Ensuite, nous avons réalisé les sous-totaux et le total général des points de fonction.

Le Tableau 40 montre le nombre des exigences (transactions et les données ou entités) et leur niveau de complexité. Le Tableau 41 donne le résultat du nombre de FP à la suite de la multiplication de chaque nombre d'exigences par le poids correspondant.

**Tableau 40 — Nombre de composants et leur niveau de complexité**

Exigences	Simple	Moyen	Complexe
<i>Intrant</i>	13	4	6
<i>Extrant</i>	4	0	0
<i>Requête</i>	3	2	1
<i>S. interne</i>	12	0	0
<i>S. externe</i>	0	0	4

**Tableau 41 — Estimation de FP non ajustés (UFP)**

UFP	FP-simple	FP-moyen	FP-complexe	FP-mixte
<i>Intrant</i>	69	92	138	91
<i>Extrant</i>	16	20	28	16
<i>Requête</i>	18	24	36	23
<i>S. interne</i>	84	120	180	84
<i>S. externe</i>	20	28	40	40
<i>Total (UFP)</i>	207	284	422	254

### Résultats avec Cocomo-I et FP

En premier lieu, nous avons introduit les données de base nécessaires à l'estimation (Exigences, niveau de complexité du traitement des exigences). Le chiffrier permet à l'estimateur de saisir le nombre des exigences et leur niveau de complexité (voir Tableau 40). Le chiffrier calcule automatiquement le total de points de fonction non ajustés (UFP en anglais) en multipliant chaque nombre d'exigences par le poids correspondant, par la suite, il calcule automatiquement l'effort de développement, le travail en milliers de lignes de code, le cout et la durée en mois pour chaque modèle (Cocomo-I et FP).

Dans le modèle FP et Cocomo-I, l'estimation de l'effort (PM pour Personne-Mois) est calculée en appliquant la formule :  $3,00 * KSLOC^{1,12}$ , de la durée (TDEV) par la formule suivante :  $2,50 * PM^{0,35}$  en mois, ainsi le nombre optimal de personnes (P) est calculé par la formule  $P = PM / TDEV$ .

**Tableau 42 — Résultat d'estimation avec FP directe**

	FP-Simple	FP-Moyen	FP-Complexe	FP-mixte
PM	41,40	56,80	84,40	50,80
TDEV	9,20	10,28	11,81	9,89
P	4,50	5,53	7,15	5,14
COST	289 800,00	397 600,00	590 800,00	355 600,00

**Tableau 43 — Résultat d'estimation avec FP et Cocomo-I**

FP et Cocomo-I	FP-Simple	FP-Moyen	FP-Complexe	FP-mixte
KSLOC	11,39	15,62	23,21	13,97
PM	45,73	65,17	101,55	57,51
TDEV	9,53	10,79	12,60	10,32
P	4,80	6,04	8,06	5,57
COST	320 120,00	456 186,00	710 846,00	402 568,00

### 5.2.3 Faisabilité de la modification de LexEVS

Cette section aborde la question découlant de l'adaptation ou de l'intégration de LexEVS avec PARS3, soit la faisabilité pratique de la modification de son code source en vue de mieux répondre aux besoins spécifiques des SSA et de PARS3 en particulier. Un premier handicap majeur est la structure même du modèle de données. Il nécessite la création d'un nouveau sous-modèle pour chaque nouvelle terminologie. À terme, cette approche n'est pas maintenable compte tenu du grand nombre de terminologies de structures différentes que le SMT requis est susceptible d'héberger.

En pratique, le modèle logique de données de LexEVS est dynamique, il faut générer de nouvelles tables lors de chaque ajout d'une terminologie. Il faut donc avoir recours (comme c'est le cas de BioPortal avec son entrepôt RDF) à la métaprogrammation, avec les mêmes conséquences : complexité, risques d'injections, performances sous-optimales.

Changer cette approche aurait un impact majeur sur la plus grande partie du code existant.

L'absence de support au traitement multilingue est un autre handicap important. En plus de nécessiter une adaptation du modèle de données, il faut généraliser l'ensemble du code des interfaces (IPM et API) de même que celui dédié à la mise en correspondance interterminologique et intraterminologique.

Au total, en ne prenant en compte que ces deux handicaps, c'est plus de 75 % du code qui doit être refactorisé, réingénierisé ou remplacé.

Finalement, d'autres fonctionnalités, peu, mal ou pas supportées par LexEVS nécessiteront le développement de nouveau code.

#### **5.2.4 Conclusion**

Bien que LexEVS permet plus de flexibilité. Alors qu'il a été initialement conçu pour accepter les fichiers terminologiques, des modules d'aide au chargeur peuvent être créés pour étendre son utilisation au chargement de modèles structurels, des modèles terminologiques et ontologiques également, mais la nécessaire modification de son modèle et donc de la plus grande partie du code nécessite beaucoup des ressources tant humaines que matérielles, ce qui emmène à conclure qu'il est plus profitable de développer un nouveau serveur, d'autant plus qu'une réécriture complète présente l'avantage de pouvoir refaire une analyse et une conception qui pourront tirer partie des avancées et des leçons obtenues par d'autres SMT.

## Conclusion

Dans ce mémoire, nous avons évalué l'adéquation des SMT couramment disponibles en regard du problème d'hétérogénéité terminologique tel qu'il se pose dans les systèmes de santé apprenants et plus particulièrement lors de la formalisation d'un projet d'accès aux données et de l'accès aux données elles-mêmes. Pour ce faire nous avons développé un ensemble de critères dérivés de cas d'utilisation et définis au chapitre 3. Une pré-évaluation a permis de réduire le nombre de candidats plausibles à trois : LexEVS, Snow Owl et BioPortal.

Ces trois SMT ont ensuite été évalués afin de déterminer lequel répond le mieux aux critères. Les trois serveurs répondent suffisamment au critère d'utilisabilité. En regard des critères relatifs au modèle de données, celui de LexEVS est accessible, mais non maintenable alors que celui de Snow Owl est inaccessible et sa maintenabilité incertaine. Quand à l'entrepôt RDF utilisé par BioPortal il reporte le problème de modélisation sur le code des fonctionnalités, code qui n'est pas disponible.

En ce qui concerne les API, celles de LexEVS et Snow Owl sont déficientes. Celle de BioPortal, dans les limites des fonctionnalités supportées par ce SMT, est manifestement plus adéquate et complète. Elle pourrait servir d'inspiration à la conception de l'API d'un nouveau SMT.

En ce qui concerne les IPM, certaines fonctionnalités de base de consultation et de recherche des terminologies font consensus et sont satisfaisantes. Cependant, elles ne couvrent pas l'ensemble des besoins découlant des SSA.

LexEVS est le seul serveur permettant de faire une liaison sémantique et structurelle. Snow Owl est le seul à offrir un modèle de sécurité avec une authentification externalisable. Aucun serveur ne supporte le multilinguisme.

Finalement, aucun des trois ne répondant suffisamment bien aux critères en l'état, il a donc fallu examiner s'il était plus profitable de développer un serveur propre ou de modifier un serveur existant afin de répondre aux exigences. Nous avons étudié la faisabilité de la modification relativement à LexEVS. L'étude a montré rapidement que le modèle de données au cœur de ce SMT était un empêchement majeur à l'évolution du code requise par la modification et l'ajout des fonctionnalités requises. Le développement d'un nouveau SMT, conforme aux critères, a donc été envisagé, puis les ressources pour ce faire, estimées à l'aide des techniques FP et Cocomo-I : le travail de 6 personnes sur une période d'un an semblerait suffisant.

Nous envisageons deux suites possibles au présent mémoire :

- ◇ Sujet à une entente avec BioPortal, examiner la faisabilité de la modification de ce SMT pour le rendre satisfaisant en regard des critères établis.
- ◇ Rédiger une spécification des exigences suffisamment complète pour en dériver une spécification d'architecture et affiner l'estimation des ressources requises au développement d'un nouveau SMT.



# Annexe A

## Modèle FP et Cocomo

### A.1 Modèle FP

#### A.1.1 Historique

L'analyse des points de fonction (Function Point Analysis (FPA) en anglais) est une méthode couramment utilisée comme technique d'estimation en génie logiciel. Définie par la première fois par Allan J. Albrecht en 1979 chez IBM, FPA, a depuis subi plusieurs modifications, principalement par l'International Function Point Users Group (IFPUG).

En termes simples, FPA est une technique utilisée pour mesurer les exigences logicielles en fonction des différentes fonctions dans lesquelles l'exigence peut être divisée. Chaque fonction est affectée de quelques points sur la base des règles FPA, puis ces points sont résumés à l'aide de la formule FPA. Le chiffre final montre le nombre total d'heures-personnes nécessaires pour atteindre l'exigence complète.

#### A.1.2 Composantes

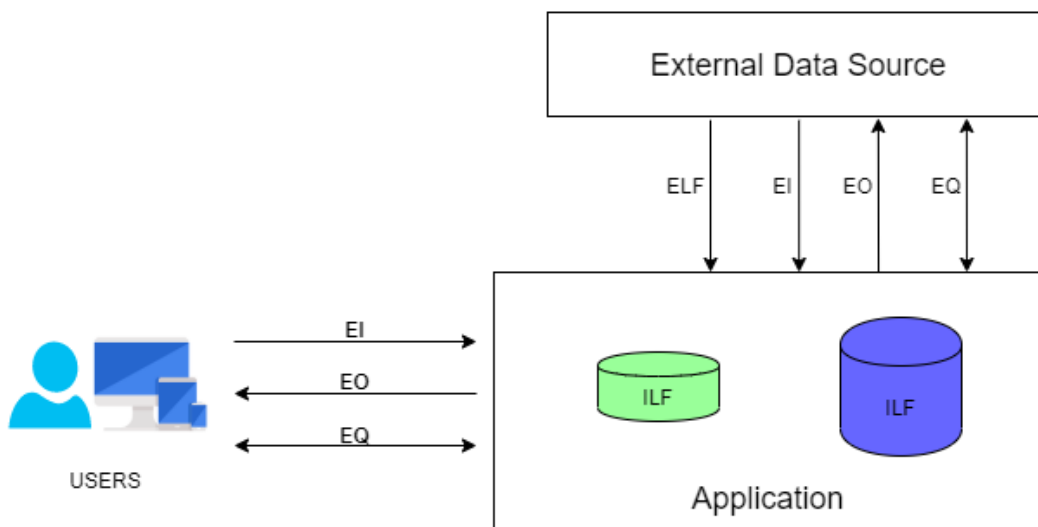
Sur la base de l'intersection des composants du système en interne et avec les utilisateurs externes, les applications, etc., ils sont classés en cinq types :

- ◇ Entrées (EI) : ce sont des données qui entrent dans l'application, initialisent ou contrôlent un traitement et font l'objet d'un traitement unique, etc.
- ◇ Sorties (EO) : il s'agit d'un processus d'envoi de données à des utilisateurs externes ou systèmes. Les données peuvent être directement extraites des fichiers de base de données ou subir un traitement au niveau du système.

- ◊ Requêtes (EQ) : ce processus comprend des composants d'entrée et de sortie. Les données sont ensuite traitées pour extraire les informations pertinentes des fichiers de base de données internes / externes.
- ◊ Fichiers Logiques Internes (ILF) : il s'agit de l'ensemble des données présentes dans le système.
- ◊ Fichiers d'Interface Externes (EIF) : il s'agit de l'ensemble de données provenant de ressources externes ou d'applications externes.

### A.1.3 Architecture

Les différents types de composants de la méthode sont illustrés par la Figure 14 ci-dessous :



**Figure 14 — Architecture du modèle des points de fonction [49]**

### A.1.4 Définitions de complexité de chaque composant

L'estimation de la taille brute de l'application (FP non ajusté) «est basée sur la complexité du traitement selon le point de vue du développeur (Complexité technique).» [50]

**Tableau 44 — Table de complexité des entrées**

	Éléments de données		
Fichiers	1-4	5-15	16+
0-1	S	S	M
2	S	M	C
3+	M	C	C

Complexité basse ou simple (S) = 3 points

Complexité moyenne (M) = 4 points

Complexité haute ou complexe (C) = 6 points

**Tableau 45 — Table de complexité des sorties**

	Éléments de données		
Fichiers	1-5	6-19	20+
0-1	S	S	M
2-3	S	M	C
4+	M	C	C

Complexité basse ou simple (S) = 4 points

Complexité moyenne (M) = 5 points

Complexité haute ou complexe (C) = 7 points

**Tableau 46 — Table de complexité des requêtes (interrogations)**

	Éléments de données		
Fichiers	1-5	6-19	20+
0-1	S	S	M
2-3	S	M	C
4+	M	C	C

Complexité basse ou simple (S) = 3 points

Complexité moyenne (M) = 4 points

Complexité haute ou complexe (C) = 6 points

**Tableau 47 — Table de complexité des fichiers internes**

Fichiers	Éléments de données		
	1-19	20-50	51+
0-1	S	S	M
2-5	S	M	C
6+	M	C	C

Complexité basse ou simple (S) = 6 points

Complexité moyenne (M) = 10 points

Complexité haute ou complexe (C) = 15 points

**Tableau 48 — Table de complexité des fichiers externes**

Fichiers	Éléments de données		
	1-19	20-50	51+
0-1	S	S	M
2-5	S	M	C
6+	M	C	C

Complexité basse ou simple (S) = 5 points

Complexité moyenne (M) = 7 points

Complexité haute ou complexe (C) = 10 points

### **A.1.5 Calcul des points de fonction**

Le calcul se fait en sommant pour chaque type de composant, les composants ayant même niveau de complexité. Chaque nombre est ensuite multiplié par le poids correspondant. On réalise ensuite les sous-totaux et le total général des points de fonction bruts.

### A.1.5.1 Calcul des points de fonction non ajustés (UFP)

**Tableau 49 — Table de calcul des points de fonction non ajustés**

Fonctions	Niveau de complexité du traitement des fonctions			Points de fonctions (4) = (1) + (2) + (3)
	Bas (1)	Moyen (2)	Élevé (3)	
Entrées	$X1*3 =$	$Y1*4=$	$Z1*6=$	
Sorties	$X2*4=$	$Y2*5=$	$Z2*7=$	
Requêtes	$X3*3=$	$Y3*4=$	$Z3*6=$	
Fichiers internes	$X4*6=$	$Y4*10=$	$Z4*15=$	
Fichiers externes	$X5*5=$	$Y5*7=$	$Z5*10=$	
	Total des points de fonction non ajustés (UFP)			(5) = $\sum$ (4)

**Notes :** X1 à X5, Y1 à Y5, Z1 à Z5 représentent le nombre de fonctions selon le niveau de complexité. Cette estimation des FP non ajustés équivaut aussi à l'estimation de l'effort non ajusté, c'est-à-dire une estimation de l'effort de développement basé sur une seule variable qui est la taille brute.

### A.1.5.2 Calcul des points de fonction ajustés

Dans cette étape le facteur d'ajustement de la valeur (VAF) est déterminé. VAF contient 14 caractéristiques générales du système (GSC) du système ou de l'application qui définit les types de caractéristiques d'application et est évalué sur une échelle de 0 à 5. La somme de tous les 14 taux GSC est calculée pour donner une valeur mathématique et est étiquetée comme influence totale du degré (TDI). Le TDI est utilisé dans le calcul du VAF et sa valeur peut varier de 0 à 35.

**Tableau 50 — Estimation des FP ajustés**

GSC	Degré d'influence (DI) (Valeur 0-1-2-3-4-5)	Interprétation DI
1. communications de données		0 : Pas d'influence 1 : Influence minimale 2: Influence modérée 3 : Influence moyenne 4 : Influence forte 5 : Influence très forte
2. Traitement réparti des données		
3. Performances		
4. Configuration très utilisée		
5. Taux de transaction		
6. Saisie de données en ligne		
7. Efficacité de l'utilisateur final		
8. Mise à jour en ligne		
9. Traitement complexe		
10. Réutilisation		
11. Installation facilité		
12. Facilité opérationnelle		
13. Facilite le changement		
14. Plusieurs sites		

$$VAF = (TDI * 0.01) + 0,65$$

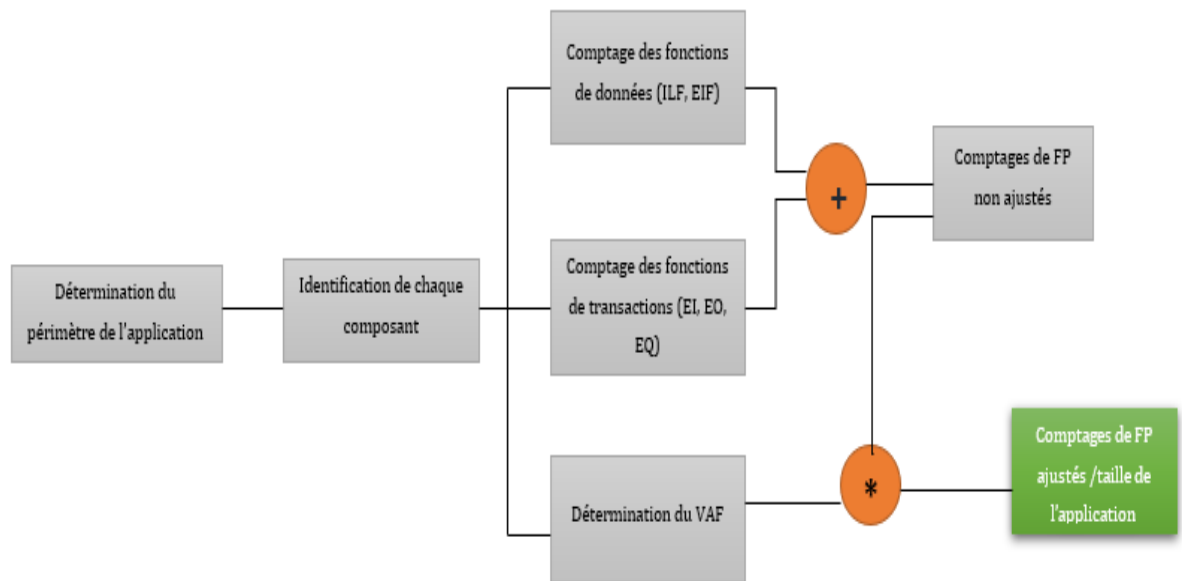
Une fois le point de fonction non ajusté et le VAF calculés, le point fonctionnel ajusté le décompte est découvert à l'aide des deux valeurs. Cela se fait à l'aide de la formule suivante :

$$FP \text{ ajusté} = FP \text{ non ajusté} * VAF$$

Le FP ajusté est ensuite multiplié par une valeur numérique, qui est l'effort basé sur la technologie.

### **A.1.6 Étapes pour compter les points de fonction**

Les étapes pour compter les points de fonction sont décrites dans la figure ci-dessous



**Figure 15 — Procédé de compte des points de fonction**

## A.2 Cocomo

### A.2.1 Description de la méthode

Cocomo (Constructive COst Model) est un modèle de régression basé sur le LOC, c'est-à-dire le nombre de lignes de code.

Il s'agit d'un modèle procédural d'estimation des coûts pour les projets logiciels et souvent utilisé comme processus de prédiction fiable des divers paramètres associés à la réalisation d'un projet, tels que la taille, l'effort, le cout, le temps et la qualité. Il a été proposé par Barry Boehm en 1970.

Cocomo-I est en fait constitué de trois modèles : modèle de base, modèle intermédiaire et modèle expert. Nous allons expliquer en détail le modèle de base que nous avons utilisé pour estimer les projets dans ce mémoire.

## A.2.2 Modèle de base

Le modèle de base est un modèle simpliste. Il estime l'effort (le nombre de mois-personne) en fonction du nombre de lignes de code, la productivité (le nombre de lignes de code par personne par mois) et un facteur d'échelle qui dépend du type de projet. Les trois types de systèmes identifiés sont :

- ◇ Autonome : organisation simple et petites équipes expérimentées (ex : système de notes dans une école).
- ◇ Couplé : entre autonome et embarqué (ex : système bancaire interactif).
- ◇ Embarqué : techniques innovantes, organisation complexe (ex : système de contrôle aérospatial).

**Tableau 51 —Types de système**

<b>Paramètres Cocomo-I</b>					
Type-système	A	E	C	F	
<i>Autonome</i>	2,40	1,05	2,50	0,38	
<i>Couplé</i>	3,00	1,12	2,50	0,35	
<i>Embarqué</i>	3,60	1,20	2,50	0,32	



## Bibliographie

- [1] D. Wassing, *Evaluation of Terminology servers for use with SNOMED CT*. Linköping, Sweden, 2020. Consulté le: nov. 30, 2020. [En ligne]. Disponible sur: <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-170792>
- [2] E. Silva Layes, M. Bondarencu, D. Machiavello, F. Frola, et M. Lemos, «Implementation of a Terminology Server with SNOMED CT in Graph Databases», *Stud Health Technol Inform*, vol. 264, p. 1584-1585, août 2019, doi: 10.3233/SHTI190546.
- [3] S. J. McBride, M. J. Lawley, H. Leroux, et S. Gibson, «Using Australian medicines terminology (AMT) and SNOMED CT-AU to better support clinical research», *Stud Health Technol Inform*, vol. 178, p. 144-149, 2012, doi: 10.3233/978-1-61499-078-9-144.
- [4] K. W. Fung, J. Xu, S. T. Rosenbloom, et J. R. Campbell, «Using SNOMED CT-encoded problems to improve ICD-10-CM coding—A randomized controlled experiment», *International Journal of Medical Informatics*, vol. 126, p. 19-25, 2019, doi: 10.1016/j.ijmedinf.2019.03.002.
- [5] J. P. Bodin, S. Albaret, et N. Melin, «CCAM et description de l'activité d'anatomocytopathologie», *Annales de Pathologie*, vol. 24, p. 56-57, 2004, doi: 10.1016/S0242-6498(04)94051-5.
- [6] R. Bravo, T. Montaner, et J. M. Pina, «Corporate brand image of financial institutions: a consumer approach», *Journal of Product & Brand Management*, vol. 21, n° 4, p. 232-245, 2012, doi: 10.1108/10610421211246649.
- [7] A. Névéol, L. F. Soualmia, M. Douyère, A. Rogozan, B. Thirion, et S. J. Darmoni, «Using CISMef MeSH "Encapsulated" terminology and a categorization algorithm for health resources», *International Journal of Medical Informatics*, vol. 73, n° 1, p. 57-64, févr. 2004, doi: 10.1016/j.ijmedinf.2003.09.004.
- [8] F. Fernandez-Llimos et T. M. Salgado, «Standardization of pharmacy practice terminology and the Medical Subject Headings (MeSH).», *Research in social & administrative pharmacy: RSAP*, vol. 17, n° 4, p. 819-820, 2021, doi: 10.1016/j.sapharm.2020.07.005.
- [9] B. E. Dixon, J. Hook, et D. J. Vreeman, «Learning From the Crowd in Terminology Mapping: The LOINC Experience», *Laboratory Medicine*, vol. 46, n° 2, p. 168-174, mai 2015, doi: 10.1309/LMWJ730SVKTUBAOJ.
- [10] Peters L., Nguyen T., Bodenreider O., et M. 16th World Congress of Medical and Health Informatics: Precision Healthcare through Informatics, «Terminology status APIs - Mapping obsolete codes to current RxNorm, SNOMED CT, and LOINC concepts», *Studies in Health Technology and Informatics*, vol. 245, 2017, doi: 10.3233/978-1-61499-830-3-1333.

- [11] D. B. Aubain, «Les référentiels sémantiques dans l'interopérabilité des systèmes d'information de santé.», 2016, [En ligne]. Disponible sur: [https://www.researchgate.net/publication/305806042\\_Les\\_referentiels\\_semantiques\\_dans\\_l'interopabilite\\_des\\_systemes\\_d'information\\_de\\_sante](https://www.researchgate.net/publication/305806042_Les_referentiels_semantiques_dans_l'interopabilite_des_systemes_d'information_de_sante)
- [12] A. Metke-Jimenez, J. Steel, D. Hansen, et M. Lawley, «Ontoserver: A syndicated terminology server», *Journal of Biomedical Semantics*, vol. 9, n° 1, 2018, doi: 10.1186/s13326-018-0191-z.
- [13] Joubert M. *et al.*, «Un serveur multi-terminologies de santé», *CEUR Workshop Proceedings*, vol. 600, p. 36-50, 2010.
- [14] G. Soula, S. Darmoni, P. L. Beux, J.-M. Renard, B. Dahamna, et M. Fieschi, «An open repositories network development for medical teaching resources.», *Studies in Health Technology and Informatics*, vol. 160, n° Pt 1, p. 610, 2010, doi: 10.3233/978-1-60750-588-4-610.
- [15] S. Darmoni, B. Thirion, S. Platel, M. Douyère, P. Mourouga, et J.-P. Leroy, «CISMeF-patient: a French counterpart to MEDLINEplus», *J Med Libr Assoc*, vol. 90, n° 2, p. 248-253, avr. 2002.
- [16] X. Aimé *et al.*, «Semantic interoperability platform for Healthcare Information Exchange», *IRBM*, vol. 36, n° 2, p. 62-69, mars 2015, doi: 10.1016/j.irbm.2015.01.003.
- [17] J. Ethier *et al.*, «The TRANSFoRM project: Experience and lessons learned regarding functional and interoperability requirements to support primary care», *Learn Health Syst*, vol. 2, n° 2, sept. 2017, doi: 10.1002/lrh2.10037.
- [18] P. Fabry, «Ontologies dans un contexte de système de santé apprenant : Nouvelles méthodologies pour faciliter leur développement et leur utilisation par des experts de domaine.», 2020.
- [19] J.-F. Ethier, «Integrating resources for translational research: a unified approach for learning health systems», p. 115, 2016.
- [20] T. Merabti, «Méthodes pour la mise en relations des terminologies médicales : contribution à l'interopérabilité sémantique Inter et Intra terminologique», These de doctorat, Rouen, 2010. Consulté le: oct. 24, 2021. [En ligne]. Disponible sur: <http://www.theses.fr/2010ROUES010>
- [21] ISO 1087:2019 (fr), «ISO 1087:2019 (fr), Travail terminologique et science de la terminologie — Vocabulaire». Consulté le: sept. 15, 2021. [En ligne]. Disponible sur: <https://www.iso.org/obp/ui/#iso:std:iso:1087:ed-2:v1:fr>
- [22] P. Zweigenbaum, «Encoder l'information médicale : des terminologies aux systèmes de représentation des connaissances», *Innovation Stratégique en Information de Santé*, vol. 2, n° 5, p. 23, 1999.

- [23] J. Grosjean, «Modélisation, réalisation et évaluation d'un portail multi-terminologique multi-discipline, multi-lingue (3M) dans le cadre de la Plateforme d'Indexation Régionale (PlaIR)», These de doctorat, Rouen, 2014. Consulté le: oct. 24, 2021. [En ligne]. Disponible sur: <http://www.theses.fr/2014ROUES028>
- [24] S. Selja, R. Alan, et S. Barry, «The Functions of Definitions in Ontologies», 2016, doi: 10.3233/978-1-61499-660-6-37.
- [25] G. Diallo, «Une Architecture à base d'Ontologies pour la Gestion Unifiées des Données Structurées et non Structurées», phdthesis, Université Joseph-Fourier - Grenoble I, 2006. Consulté le: oct. 24, 2021. [En ligne]. Disponible sur: <https://tel.archives-ouvertes.fr/tel-00221392>
- [26] A. Moreno-Conde, T. Austin, J. Moreno-Conde, C. L. Parra-Calderón, et D. Kalra, «Evaluation of clinical information modeling tools», *J Am Med Inform Assoc*, vol. 23, n° 6, p. 1127-1135, nov. 2016, doi: 10.1093/jamia/ocw018.
- [27] D. D. Ahmed-Diouf *et al.*, «Recherche d'information multiterminologique en contexte: étude préliminaire», 2009, p. 101-112. doi: 10.1007/978-2-287-99305-3\_10.
- [28] L. Traore *et al.*, «Plateforme d'interopérabilité sémantique gérant les terminologies d'interface au sein d'un espace de partage», mai 2014, p. 75. Consulté le: oct. 24, 2021. [En ligne]. Disponible sur: <https://hal.inria.fr/hal-01015196>
- [29] N. Griffon, «Modélisation, création et évaluation de ux de terminologies et de terminologies d'interface: application à la production d'examens complémentaires de biologie et d'imagerie médicale.», p. 165, 2013.
- [30] Agence Française de la santé numérique, «Référentiels d'interopérabilité sémantique : mise en œuvre de terminologies de référence pour le secteur santé-social en France», France, 2019. [En ligne]. Disponible sur: [https://esante.gouv.fr/sites/default/files/media\\_entity/documents/asip\\_termino\\_rapport\\_phase\\_4\\_vf1.3.1\\_vf.pdf](https://esante.gouv.fr/sites/default/files/media_entity/documents/asip_termino_rapport_phase_4_vf1.3.1_vf.pdf)
- [31] Robert M. Colomb et Mohammad Nazir Ahmad, «Overview of Ontology Servers Research», *Webology*, vol. 4, n° 2, p. 43.
- [32] Grosjean J. *et al.*, «Integrating the human phenotype ontology into HeTOP terminology-ontology server», *Studies in Health Technology and Informatics*, vol. 192, n° 1-2, 2013, doi: 10.3233/978-1-61499-289-9-961.
- [33] J.-F. Ethier *et al.*, «A unified structural/terminological interoperability framework based on lexEVS: Application to TRANSFoRm», *Journal of the American Medical Informatics Association*, vol. 20, n° 5, p. 986-994, 2013, doi: 10.1136/amiajnl-2012-001312.
- [34] B. Balázs, P. Roland, et K. Tibor, «Implementing reusable software components for SNOMED CT diagram and expression concept representations», *e-Health – For Continuity of Care*, p. 1028-1032, 2014, doi: 10.3233/978-1-61499-432-9-1028.

- [35] Patricia L. Whetzel *et al.*, «BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications», *Nucleic Acids Research*, vol. 39, n° 2, p. W541-W545, 2011, doi: 10.1093/nar/gkr469.
- [36] M. Salvadores, P. R. Alexander, M. A. Musen, et N. F. Noy, «BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF», *Semant Web*, vol. 4, n° 3, p. 277-284, 2013.
- [37] N. F. Noy *et al.*, «BioPortal: ontologies and integrated data resources at the click of a mouse», *Nucleic Acids Res.*, vol. 37, n° Web Server issue, p. W170-173, juill. 2009, doi: 10.1093/nar/gkp440.
- [38] ISO 9241-11:2018 (fr), «ISO 9241-11:2018 (fr), Ergonomie de l'interaction homme-système — Partie 11: Utilisabilité — Définitions et concepts». 2018. Consulté le: avr. 06, 2021. [En ligne]. Disponible sur: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:fr>
- [39] J. Nielsen, *Usability engineering*, Nachdr. Amsterdam: Kaufmann, 2010.
- [40] B2i Healthcare, «Snow Owl Server Documentation», p. 67.
- [41] M. Martínez-Romero, C. Jonquet, M. J. O'Connor, J. Graybeal, A. Pazos, et M. A. Musen, «NCBO Ontology Recommender 2.0: an enhanced approach for biomedical ontology recommendation», *J Biomed Semant*, vol. 8, n° 1, p. 21, déc. 2017, doi: 10.1186/s13326-017-0128-y.
- [42] J. Pathak, L. Peters, C. G. Chute, et O. Bodenreider, «Comparing and evaluating terminology services application programming interfaces: RxNav, UMLS and LexBIG», *Journal of the American Medical Informatics Association*, vol. 17, n° 6, p. 714-719, nov. 2010, doi: 10.1136/jamia.2009.001149.
- [43] J. Pathak, H. R. Solbrig, J. D. Buntrock, T. M. Johnson, et C. G. Chute, «LexGrid: A Framework for Representing, Storing, and Querying Biomedical Terminologies from Simple to Sublime», *Journal of the American Medical Informatics Association*, vol. 16, n° 3, p. 305-315, mai 2009, doi: 10.1197/jamia.M3006.
- [44] Y. He, «Ontology-Based Vaccine and Drug Adverse Event Representation and Theory-Guided Systematic Causal Network Analysis Toward Integrative Pharmacovigilance Research», *Current Pharmacology Reports*, vol. 2, n° 3, p. 113-128, 2016, doi: 10.1007/s40495-016-0055-0.
- [45] W. Ceusters, B. Smith, et L. Goldberg, «A terminological and ontological analysis of the NCI Thesaurus», *Methods Inf Med*, vol. 44, n° 4, p. 498-507, 2005.
- [46] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, et R. Selby, «Cost models for future software life cycle processes: COCOMO 2.0», *Ann Software Eng*, vol. 1, n° 1, p. 57-94, déc. 1995, doi: 10.1007/BF02249046.

- [47] R. Dumke, R. Neumann, A. Schmietendorf, et C. Wille, «Empirical-Based Extension of the COSMIC FP Method», in *2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement*, oct. 2014, p. 5-10. doi: 10.1109/IWSM.Mensura.2014.20.
- [48] A. Z. Abualkishik et L. Lavazza, «IFPUG Function Points to COSMIC Function Points convertibility: A fine-grained statistical approach», *Information and Software Technology*, vol. 97, p. 179-191, 2018, doi: 10.1016/j.infsof.2018.01.012.
- [49] ARC Optimizer, «Analyse des points de fonction - Introduction et principes de base», *Blog ARC Optimizer*, janv. 14, 2020. <https://blog.arcoptimizer.com/analyse-des-points-de-fonction-introduction-et-principes-de-base> (consulté le oct. 25, 2021).
- [50] O. Ktata, «Estimation de projets web: application et analyse de fiabilité des modèles COCOMO II et WebMo», 2007, [En ligne]. Disponible sur: <https://archipel.uqam.ca/3264/1/M9720.pdf>